

# Optimización de reactores utilizando el simulador de procesos orientado a ecuaciones EMSO



Grado en Ingeniería Química

## Trabajo Fin de Grado

Autor:

Valentina Montes Giraldo

Tutores:

Rubén Ruiz Femenia | Juan Javaloyes Anton

Julio 2021



Universitat d'Alacant  
Universidad de Alicante

## Resumen

En la actualidad es importante contar con sistemas informáticos competentes capaces de abarcar un amplio abanico de problemas, operar muchas ecuaciones simultáneamente, optimizando objetivos que en muchos casos son opuestos, y al mismo tiempo que estos sistemas informáticos sean capaces de dar una respuesta inmediata o en el menor tiempo posible.

Cada vez se escucha más el concepto de sostenibilidad, la capacidad de aprovechar los recursos de la mejor manera, por lo que ahora, más que nunca, es necesario un sistema competente que sea capaz de gestionar todos estos requisitos y los futuros retos que se puedan presentar debido, principalmente, a la escasez de recursos. En este trabajo se pretende hacer una introducción general al simulador EMSO, un simulador orientado a ecuaciones, que es muy útil en el ámbito académico gracias a su versatilidad, facilidad de uso, y que no requiere una gran capacidad de almacenamiento, además es gratuito para las instituciones académicas.

Con el objetivo de dar a conocer el programa y una mejor comprensión de su funcionamiento se realizarán varios casos de estudio orientados, principalmente, a la simulación y optimización. Finalmente, se expondrán las ventajas e inconvenientes del simulador de procesos EMSO con respecto a otros simuladores que hay en el mercado.

## Palabras clave

EMSO

Simulador orientado a ecuaciones

Simulación procesos químicos

Optimización de procesos químicos

# ÍNDICE

<b>1. Introducción.....</b>	<b>3</b>
<b>2. Antecedentes.....</b>	<b>5</b>
<b>3. Objetivos .....</b>	<b>6</b>
<b>4. Metodología .....</b>	<b>6</b>
<b>5. Desarrollo.....</b>	<b>7</b>
5.1. Introducción al manejo del simulador de procesos EMSO.....	7
5.2. Conceptos básicos de modelado.....	9
5.3. Unidades de medida en EMSO .....	11
5.4. Optimización en EMSO .....	11
5.5. Resultados en EMSO .....	12
5.6. Casos de estudio.....	13
5.6.1. Simulación de tres tanques en serie. ....	13
5.6.2. Optimización de tres reactores continuos de tanque agitado en serie .....	16
5.6.3. Planificación de la producción .....	19
5.6.4. Optimización de la producción de amoniaco .....	24
5.6.4.1. Análisis económico de la producción de amoniaco .....	29
5.6.5. Simulación de la producción de anhídrido acético.....	32
<b>6. Resultados.....</b>	<b>37</b>
6.1. Resultados de los casos de estudio .....	37
6.1.1. Simulación de tres tanques en serie .....	37
6.1.2. Optimización de tres reactores continuos de tanque agitado en serie .....	41
6.1.3. Planificación de la producción .....	41
6.1.4. Optimización de la producción de amoniaco .....	42
6.1.4.1. Resultados del análisis económico .....	44
6.1.5. Simulación de la producción de anhídrido acético.....	45
6.2. Comparación resultados en Matlab y EMSO .....	47
6.4. Comparación resultados en GAMS y EMSO .....	51
6.5. Comparación resultados en Matlab, Aspen Hysys y EMSO .....	53
<b>7. Conclusiones.....</b>	<b>59</b>
<b>8. Apéndices y anexos .....</b>	<b>60</b>
Apéndice I: Tablas resumen de las referencias más utilizadas en EMSO. ....	60
Anexo I: Scripts de los modelos de equipos utilizados en la simulación y optimización de la producción de amoniaco. ....	67
Anexo II: Simulación de la producción de anhídrido acético en el simulador Aspen Hysys. ....	69
<b>9. Bibliografía.....</b>	<b>72</b>

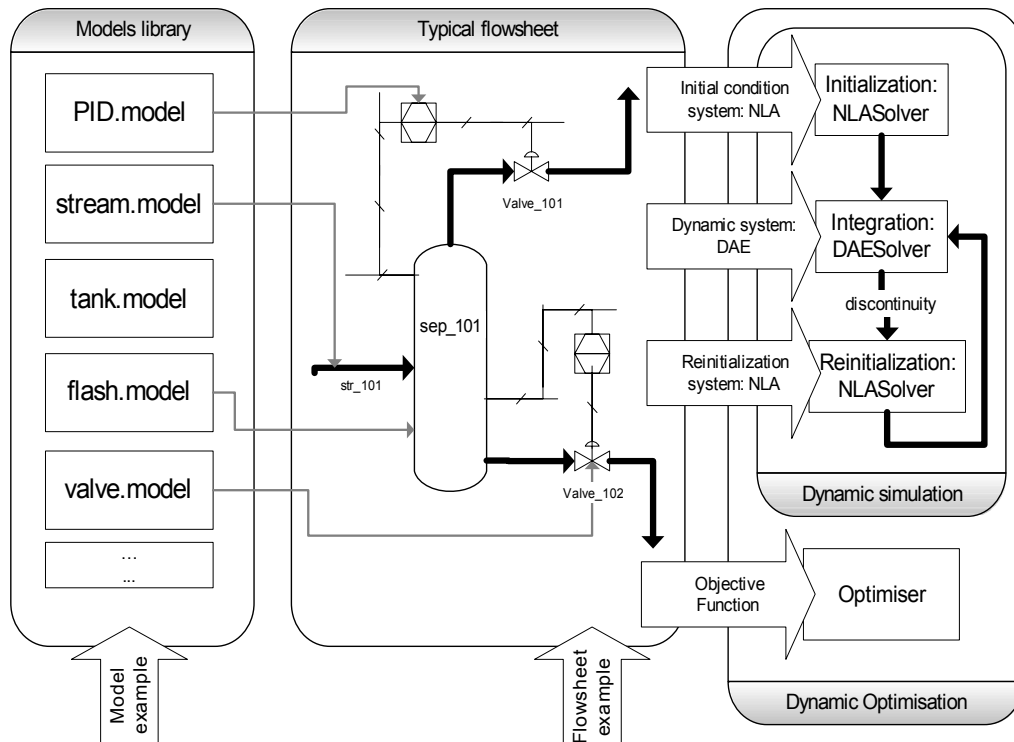
## 1. Introducción

Es importante contar con simuladores competentes en el mercado para llevar a cabo procesos estrechamente integrados con varios balances de materia y energía. Con el objetivo de tener una mejor comprensión de los procesos más complejos, la optimización y construcción de procesos, para la construcción de prototipos virtuales, verificación de diseño, etc. Se puede obtener información de un proceso a partir de experimentos, pero estos suelen llevar mucho tiempo, suelen ser costos y peligrosos. En la actualidad es más necesario que nunca contar con los simuladores adecuados debido a que no solo se tiene en cuenta el factor económico, además, son importantes otros factores como la huella de carbono, el impacto que tiene la industria en la sociedad, el tipo de contaminación que genera, cumplir con las normas y requisitos exigidos por los gobiernos.

En el mercado existen muchos simuladores de procesos. Por un lado, se pueden encontrar los simuladores de procesos secuenciales-modulares donde los cálculos del equipo se realizan de forma secuencial, el resultado de dejar un bloque se pasa como entrada al siguiente bloque, iterando a la solución de reciclaje. Existen simuladores secuenciales-modulares en el mercado como Aspen HYSYS, CHEMCAD o el simulador COCO.

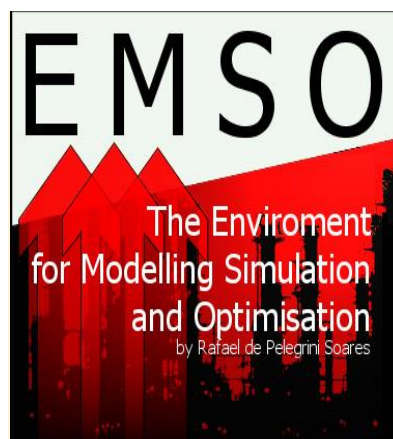
Por otro lado, están los simuladores de procesos orientados a ecuaciones donde usando el mismo conjunto de plantillas, es posible realizar infinidad de tareas diferentes tales como simulaciones estacionarias, simulaciones dinámicas, estimación de parámetros, reconciliación de datos, optimización estacionaria, optimización dinámica, además las plantillas se pueden ver, modificar o ampliar fácilmente. En la figura 1 se puede observar un ejemplo descriptivo de cómo funcionan los simuladores orientados a ecuaciones, que constan de un flowsheet formado de varios modelos de equipos que en su conjunto forman la simulación.

En el mercado se pueden encontrar simuladores orientados a ecuaciones como gPROMS y EMSO. Actualmente, los simuladores modulares dominan el mercado, pero cada vez son más necesarios los simuladores orientados a ecuaciones ya que se presentan problemas cada vez más de mayor tamaño que implican, muchas veces, más de 100.000 ecuaciones, y que requieren de su optimización.



**Figura 1.** Simulador de procesos orientado a ecuaciones (Dr. Rafael de Pelegrini Soares, 2001).

El simulador de procesos EMSO cuyas siglas significan Environment for Modeling, Simulation and Optimization (Entorno para Modelado, Simulación y Optimización), fue desarrollado por el Dr. Rafael de Pelegrini Soares a partir del año 2001 en la Universidad Federal de Rio Grande do Sul, en Brasil. Es un simulador de procesos orientado a ecuaciones con una interfaz gráfica para modelar procesos complejos dinámicos o de estado estacionario escrito en el lenguaje de programación C++, los modelos están escritos en un lenguaje de modelado, y está disponible tanto para el sistema operativo Windows, como para el sistema operativo Linux.



**Figura 2.** Logo de EMSO, Environment for Modeling, Simulation and Optimization.

En 2005 inició el proyecto ALSOC para mejorar el sistema que desarrolla, mantiene y distribuye este software orientado a objetos (Soares, ALSOC, s.f.). Los modelos prediseñados están disponibles en EMSO Modeling Library (EML), la base de datos del programa donde se encuentran los modelos de los equipos más utilizados en la industria química. Los nuevos modelos se pueden escribir en el lenguaje de modelado EMSO o se pueden incluir modelos codificados en C, C++ o Fortran en el entorno de simulación.

EMSO fue creado, principalmente, debido a que el uso de simuladores en la industria brasileña es limitado por falta de formación de los ingenieros de procesos, las complejidades en la construcción y análisis de modelos, y el alto costo para renovar las licencias anuales.

## 2. Antecedentes

Las aplicaciones informáticas se han convertido en herramientas imprescindibles en todas las áreas de la ingeniería. Gracias al diseño asistido por ordenador se pueden estimar los costes de un gran proyecto y poder considerar si es o no rentable, diseñar, y poder tener una visión más real del proyecto final sin necesidad de tener que hacer una gran inversión previamente.

El sector de la industria química no es una excepción, gracias a los simuladores se pueden tener en cuenta muchos problemas de diseño, antes incluso de la ejecución del proyecto y en el caso de trabajar con sustancias tóxicas o peligrosas poder prevenir muchos accidentes.

Igualmente, hoy en día existe una preocupación debido al gran crecimiento de la población mundial y la demanda de materiales en general. Por lo que cada vez se tiene más en cuenta el concepto de optimización de sistemas para aprovechar de una manera más eficiente todos los recursos presentes en el planeta. Por ese motivo los simuladores orientados a ecuaciones, como EMSO, son muy útiles gracias a su versatilidad para optimizar las ecuaciones que simulan procesos productivos.

Existe una gran demanda en el sector de la industria química, así como en las universidades y distintos centros académicos, de simuladores de procesos, y existe la necesidad de implementar sistemas cada vez más modernos con resultados más precisos. El simulador de procesos EMSO es una gran herramienta para las universidades, gracias a su sencillez, versatilidad, a que es bastante intuitivo, y además es gratuito en el ámbito académico.

### 3. Objetivos

Los objetivos del presente trabajo fin de grado es ofrecer una introducción general sobre el software EMSO, un simulador de procesos orientado a ecuaciones, dando a conocer el funcionamiento básico del programa mediante una serie de casos de estudio propuestos, al mismo tiempo que se analizan las ventajas e inconvenientes que suponen el uso de un simulador orientado a ecuaciones frente a un simulador modular, que son los simuladores más utilizados en la actualidad.

Principalmente, se explicará de forma resumida los aspectos más relevantes del programa, se plantearán diferentes casos de estudio de optimización de procesos haciendo uso de las distintas herramientas del programa con la finalidad de dar a conocer la estructura de los modelos y diagramas de flujo. Asimismo, para entender mejor las distintas posibilidades que ofrece el programa se hará uso de la base de datos de EMSO y se comparará con la introducción manual de las ecuaciones de los modelos.

El objetivo final es que el lector sea capaz de aprender de forma independiente, mediante el presente proyecto, el manejo del simulador de procesos EMSO obteniendo un nivel en el que el usuario sea capaz de simular y optimizar los procesos más comunes en el campo de la ingeniería química.

### 4. Metodología

Para aprender de forma autodidacta el manejo del simulador EMSO, se ha optado por la siguiente metodología.

Primeramente, se ha realizado una búsqueda de la información más relevante centrada, principalmente, en el manual de uso de EMSO, (Soares, EMSO Manual, 2002-2004). Además, se han utilizado diferentes fuentes de investigación en las que se hace uso del programa EMSO para la simulación y optimización de procesos químicos (João Paulo Henrique<sup>1</sup> | Ruy de Sousa Jr.1, 2017).

También, se abordan las cuestiones más relevantes que se deben tener en cuenta del simulador de procesos, para facilitar al lector el aprendizaje correcto del mismo. Para ello igualmente se puede hacer uso de las tablas “Quick references” presentes en el anexo I que se han presentado en el anexo para facilidad de uso, pero que se encuentran en la página web de EMSO<sup>1</sup>.

Con el objetivo de aprender y poner a prueba las competencias adquiridas, se llevan a cabo diferentes casos de estudio que aumentan de dificultad gradualmente (João Paulo Henrique<sup>1</sup> | Ruy de Sousa Jr.1, 2017). Teniendo la posibilidad de ver distintos ejemplos de optimización de procesos, tanto de sistemas simples como complejos.

Por último, una vez se tiene una idea general del uso del simulador EMSO, se lleva a cabo una comparación frente a otros simuladores comerciales, comparando las distintas ventajas o inconvenientes que presenta el programa.

---

<sup>1</sup> <https://www.enq.ufrgs.br/trac/alsoc/wiki/EMSO>

## 5. Desarrollo

### 5.1. Introducción al manejo del simulador de procesos EMSO

La instalación del programa EMSO se realiza mediante la pagina web (UFRGS, 2002) donde se puede encontrar tanto el programa como información adicional de fuentes relacionadas, publicaciones, guías y problemas resueltos.

El simulador de procesos EMSO es un simulador basado en ecuaciones por lo que necesita contar con complementos (plugins) ya que hay algunos casos en los que el uso de ecuaciones es poco conveniente. El complemento es un mecanismo de interfaz que permite al usuario cargar servicios de cálculo proporcionados por software de terceros dentro de EMSO. Los casos típicos en los que se recomienda el uso de complementos son cuando se requieren los siguientes cálculos:

- Propiedad basada en correlaciones.
- Propiedades termo físicas.
- Cálculos de dinámica de fluidos computacional para geometrías complejas.

Además, el programa cuenta con una amplia base de datos (eml) de los equipos más utilizados en la industria química, como se puede observar en la siguiente figura dentro de la ventana «Explorer» (recuadro rojo indicado en la imagen). En la figura 4 se puede observar un esquema detallado de los modelos existentes en la base de datos eml de los distintos equipos más utilizados en la industria química.

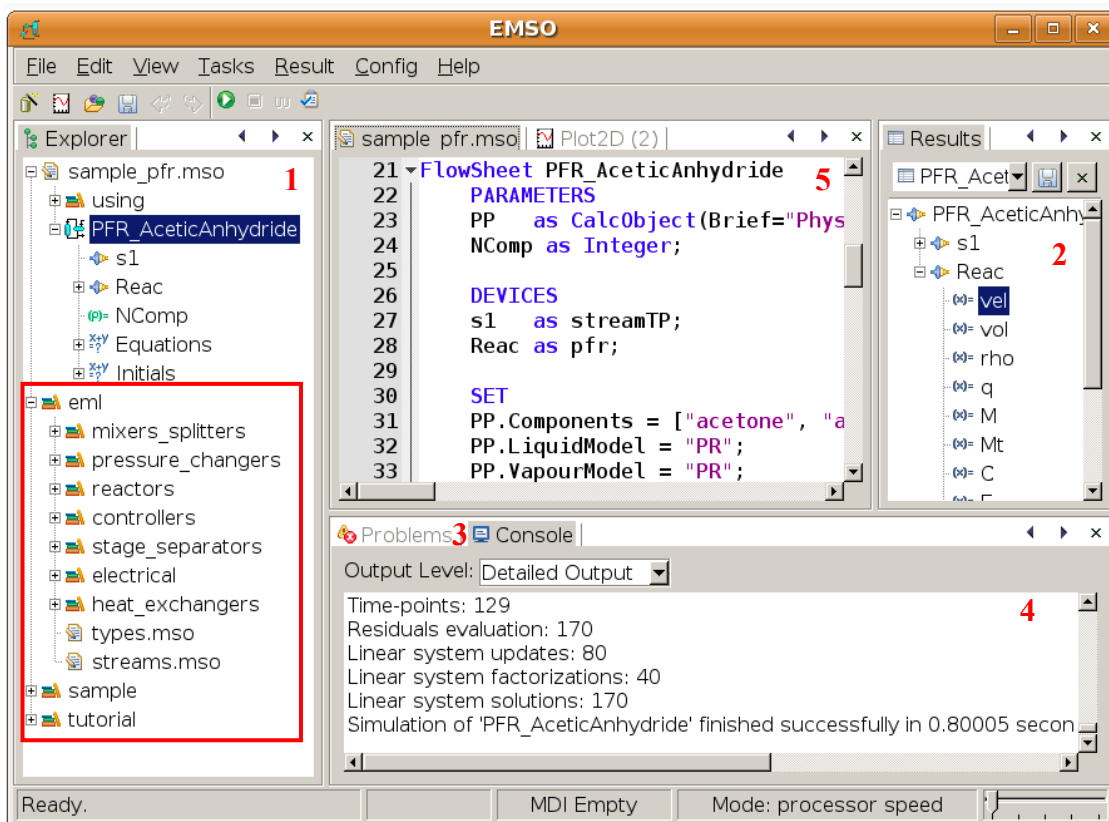


Figura 3. Interfaz gráfica de EMSO, con la numeración de las distintas ventanas.



La interfaz gráfica se divide en, según la figura 3:

1. Explorador: Muestra las bibliotecas de modelos disponibles y los archivos cargados actualmente y su contenido (Modelos y FlowSheets).
2. Resultados: Vista general de los resultados obtenidos.
3. Problemas: Enumera automáticamente todos los errores y advertencias encontrados.
4. Consola: Al ejecutar tareas, todos los mensajes se envían a la consola.
5. El panel de Interfaz de varios documentos: Este panel se encarga de mostrar los archivos abiertos y editarlos, los gráficos de resultados, etc.

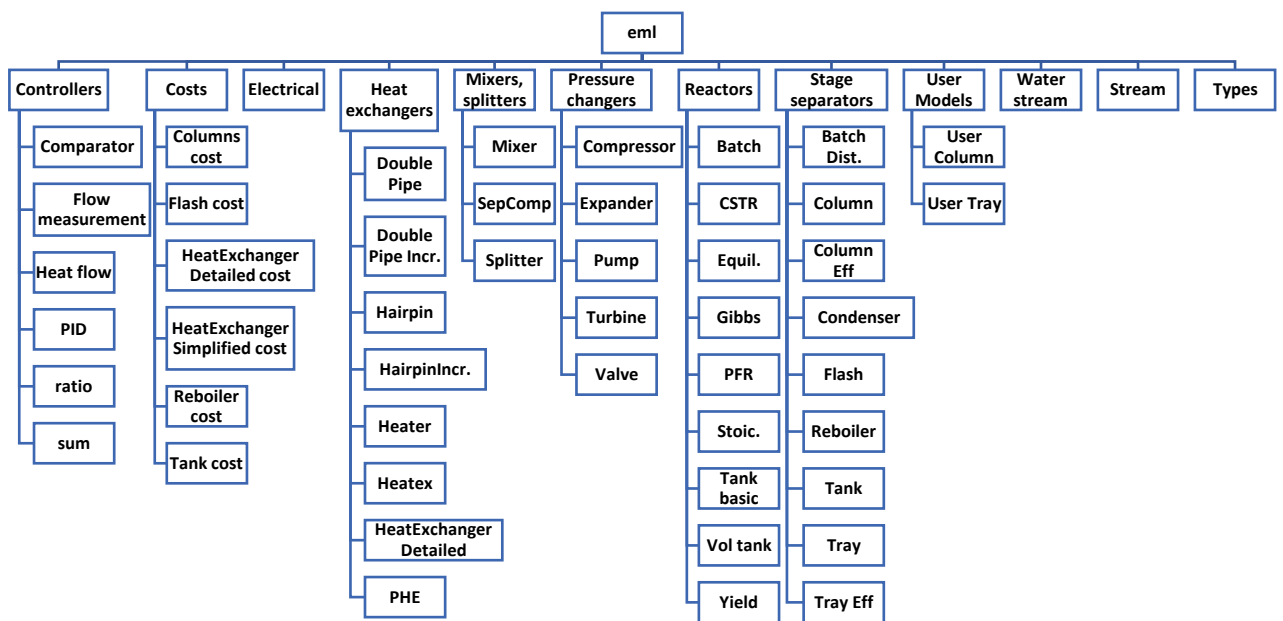


Figura 4. Resumen de la base de datos de EMSO (eml).

## 5.2. Conceptos básicos de modelado

Para simular un proceso en EMSO, es importante diferenciar las siguientes partes: Model, Flowsheet y Optimization.

El Model es el conjunto de modelos matemáticos que representa un equipo en concreto. Por ejemplo, se puede encontrar modelos de compresores centrífugos, intercambiadores de calor, bombas, turbinas, reactores...

El flowsheet es el problema en estudio, compuesto por un conjunto de modelos y dispositivos. Donde se especifican las conexiones entre los diferentes equipos que forman el proceso.

La Optimización (Optimization) no sólo pretende simular el proceso, sino que además intenta encontrar la mejor solución.

Una ventaja añadida del simulador EMSO es la diferenciación simbólica y automática que es capaz de diferenciar los distintos comandos, números y formatos mediante colores que se incorporan automáticamente al escribir, como se puede observar en los scripts presentes en la parte inferior de la figura 5.

Para entender mejor cómo se correlacionan los diferentes modelos dentro del flowsheet, se puede observar en la siguiente figura 5, un ejemplo visual de un sistema básico de separación de compuestos.

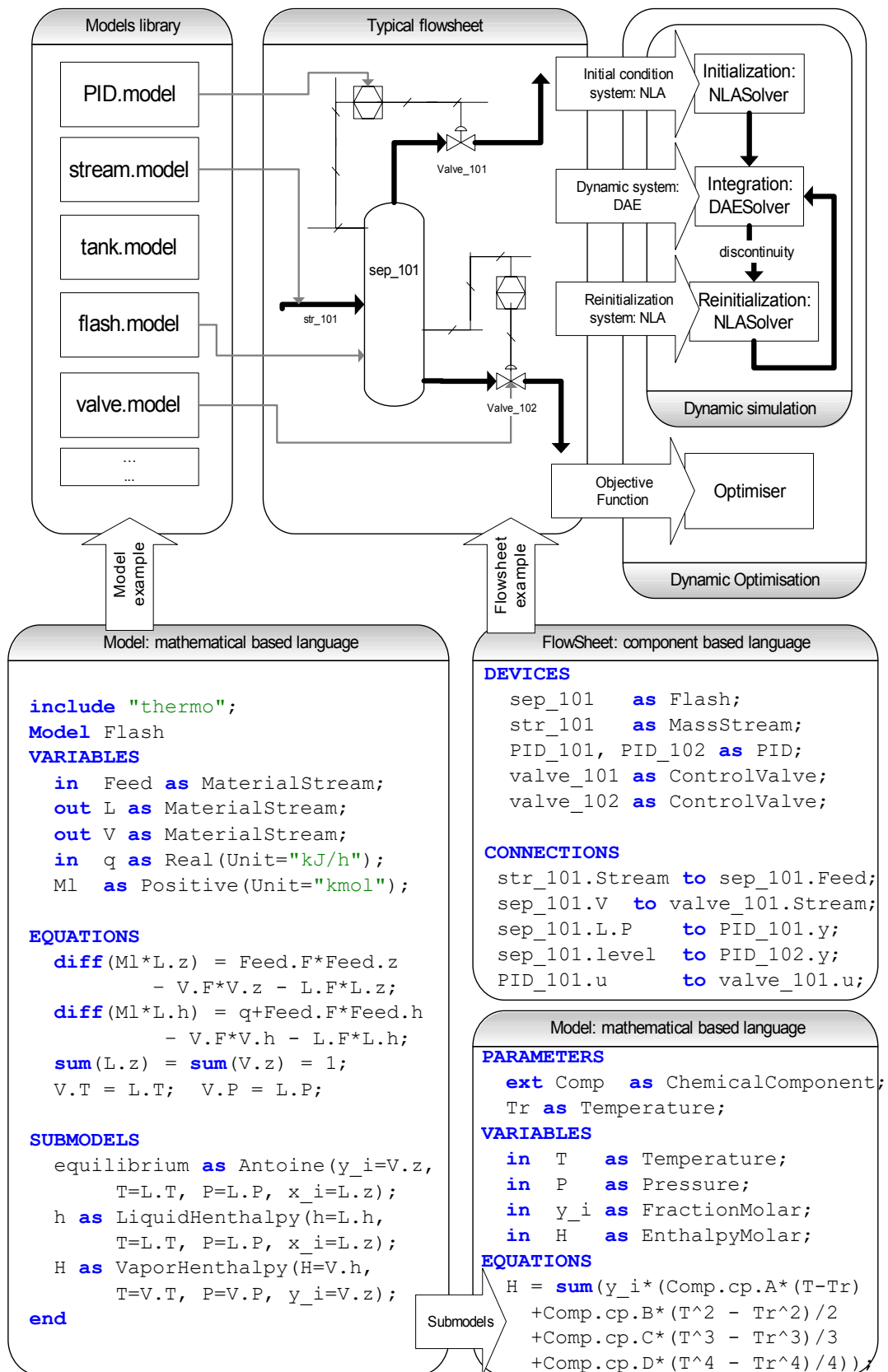


Figura 5. Ejemplo visual de correlación de los modelos matemáticos en EMSO.

En la figura 5 se puede observar un flowsheet compuesto por diferentes equipos: Separador flash, intercambiador de calor y válvula. Cada equipo corresponde a un modelo en concreto, y todos los modelos en su conjunto forman el flowsheet. En la parte inferior de la figura se puede observar un script real sobre un flowsheet en EMSO compuesto de varios modelos de equipos, en el flowsheet se especifican las conexiones entre equipos y las especificaciones del problema.

### 5.3. Unidades de medida en EMSO

El simulador de procesos EMSO reconoce una amplia variedad de unidades de medida, que se pueden encontrar en las tablas “Quick references” presentes en el anexo I. Algunas de las más comunes se pueden observar en la figura 6.

#### Fundamental Units

m	length in meters
kg	mass in kilogram
s	time in seconds
K	temperature in Kelvin
A	electric current in Ampere
mol	the amount of substance in mole
cd	the luminous intensity in Candela
rad	angle measure in radian
US\$	money in dollar (USA)

*Figura 6. Unidades fundamentales en EMSO.*

Es importante resaltar que es necesario poner todas las unidades en las que se está trabajando a la hora de simular u optimizar un proceso en EMSO. Además, una gran ventaja del simulador EMSO es que no exige que las unidades sean equivalentes. Por ejemplo, si en un modelo se fija la presión en atmósferas y luego en el flowsheet se especifica la presión en Pascales, no supondría ningún problema.

### 5.4. Optimización en EMSO

Actualmente solo es posible optimizar problemas en estado estacionario, aunque se está trabajando en la optimización dinámica y muy pronto se podrán abordar este tipo de problemas con EMSO. Se pueden abordar problemas de programación no lineal NLP (non Linear Programing), con NLPSolver, y de ecuaciones no lineales MINLP (Mixed Integer non Linear Programing), con MINLPSolver. En la figura 7 se puede observar un ejemplo de optimización no lineal NLP en EMSO. El NLPSolver utiliza el método de Newton con Diferenciación Automática. La Diferenciación Automática (AD) se basa en el concepto matemático de álgebras diferenciales, que disminuye la carga de trabajo que supone linealizar sistemas no lineales al solver NLP y a la vez mejora la eficiencia de los equipos informáticos (E. Tijskens a, 2002).

```

OPTIONS
Dynamic = false;
NLPsolveNLA = true;
# FeasiblePath = true;
NLPsolver(#File = "complex",
           #File = "optpp_emso",
           File = "ipopt_emso",
           MaxIterations = 300,
           RelativeAccuracy = 1e-6);
end

```

Figura 7. Ejemplo de una optimización NLP.

Por otro lado, EMSO tiene la opción de incorporar nuevos solver externos como NLA (non Linear Algebraic), DAE (Differential-Algebraic Equation), NLP o MINLP. Las rutinas externas y los solvers se implementan como bibliotecas compartidas utilizando plantillas proporcionadas dentro de EMSO.

## 5.5. Resultados en EMSO

EMSO es un simulador muy versátil que se puede utilizar junto con otros programas tales como: Matlab o Excel. Los resultados pueden presentarse en la ventana «Results», o pueden guardarse en archivos Matlab o Excel, como se muestra en la figura 8.

DeviceName	StreamName	FEED	C101	R101	F101
		Outlet	Inlet	Outlet	Inlet
F	'kmol/h'	907,1850	908,3650	1028,9200	195,7290
T	'K'	300,1500	300,1070	701,6860	1766,6600
P	'atm'	10,0000	10,0000	200,0000	200,0000
h	'kJ/kmol'	-1,9065	-3,2722	12146,5000	55758,9000
v		1,0000	1,0000	1,0000	1,0000
z(1) H2		0,7400	0,7393	0,7349	0,5551
z(2) N2		0,2400	0,2399	0,2378	0,1762
z(3) Ar		0,0100	0,0100	0,0108	0,0133
z(4) CH3		0,0100	0,0101	0,0108	0,0133
z(5) NH3		0,0000	0,0006	0,0058	0,2420
Production	kmol/h				
Pérdida NH3	kmol/h				

Figura 8. Resultados de EMSO en archivo Excel.

Como se podrá observar más adelante en el ejemplo de la simulación de tres tanques en serie, EMSO tiene la capacidad de crear gráficos con los resultados de las simulaciones. Muy útiles a la hora de comparar compuestos u observar el comportamiento de una sustancia en una reacción.

## 5.6. Casos de estudio

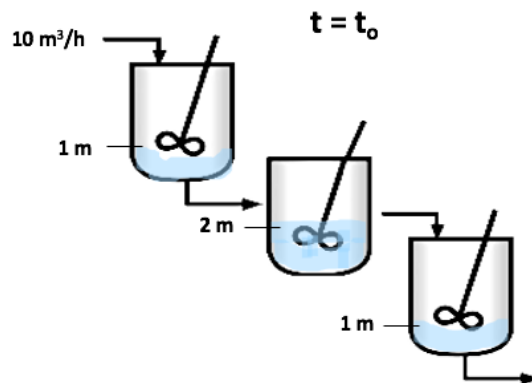
Se han planteado una serie de casos de estudio para entender mejor el funcionamiento del simulador EMSO mediante la práctica.

Aumentando gradualmente de dificultad, se han propuesto un total de cuatro casos de estudio:

- 5.6.1. Simulación de tres tanques en serie.
- 5.6.2. Optimización de tres reactores continuos de tanque agitado en serie.
- 5.6.3. Planificación de la producción.
- 5.6.4. Optimización de la producción de amoniaco.
- 5.6.5. Simulación de la producción de anhídrido acético.

### 5.6.1. Simulación de tres tanques en serie.

Se plantea el siguiente problema: Tres reactores en serie cuyo caudal de entrada es de  $10 \text{ m}^3/\text{h}$  y cuyas alturas iniciales, dentro de cada tanque, son 1, 2 y 1 m respectivamente para los tanques 1, 2 y 3. Se desea conocer el caudal de salida en cada reactor y la variación de nivel en cada tanque en función del tiempo.



*Figura 9. Caso inicial del problema tres tanques en serie.*

En la figura 10 se puede observar el modelo que proporciona el simulador EMSO, al hacer referencia al modelo tank\_simplified. Como se puede observar, el modelo incluye los balances de materia, la relación entre la altura del nivel o la ecuación de velocidad.

```

Model tank_simplified

PARAMETERS
  pi as Real (Brief="pi number",Default=3.141592);
  k as Real (Brief="Valve Constant", Unit = 'm^2.5/h', Default=4);
  Dh as Real (Brief="Hydraulic diameter", Unit = 'm', Default=2);
  A as Real (Brief="Tank area", Unit = 'm^2');

SET
  pi = 3.141592;
  A = (pi*Dh^2)/4;

VARIABLES
  V as Real (Brief="Tank Volume", Lower = 0, Unit='m^3');
  Level as Real (Brief="Tank level", Lower = 0, Unit='m');
  in Fin as Real (Brief="Input flow", Lower = 0, Unit='m^3/h');
  out Fout as Real (Brief="Output flow",Lower = 0, Unit='m^3/h');

EQUATIONS
  "Mass balance"
    diff(V) = Fin - Fout;

  "Valve equation"
    Fout = k*sqrt(Level);

  "Tank volume"
    V = A*Level;

end

```

Figura 10. Modelo tank\_simplified, que se encuentra en la base datos del programa (eml).

Se plantea el problema haciendo referencia al modelo de tanque simplificado (“tank\_simplified”) que se encuentra en el programa tank, dentro de la carpeta stage\_separators, por eso es necesario añadir el comando using “stage\_separators/tank”, todo ello dentro de la base de datos de EMSO (eml). Para utilizar este modelo es necesario especificar la corriente de entrada y los niveles iniciales de los reactores.

Las ecuaciones que utiliza el modelo tank\_simplified son las siguientes:

$$dV = F_{in} - F_{out} \quad (1)$$

$$F_{out} = k\sqrt{L} \quad (2)$$

$$V = A \cdot L \quad (3)$$

Donde V es el volumen del reactor, F el caudal de entrada o salida, k la constante de la velocidad de reacción, A en área y L la altura de nivel dentro del tanque.

```

using "stage_separators/tank";

FlowSheet ThreeTank
  VARIABLES
    Feed as flow_vol;

  DEVICES
    Tank1 as tank_simplified;
    Tank2 as tank_simplified;
    Tank3 as tank_simplified;

  CONNECTIONS
    Feed to Tank1.Fin;
    Tank1.Fout to Tank2.Fin;
    Tank2.Fout to Tank3.Fin;

  SPECIFY
    Feed = 10 * 'm^3/h';

  INITIAL
    Tank1.Level = 1 * 'm';
    Tank2.Level = 2 * 'm';
    Tank3.Level = 1 * 'm';

end

```

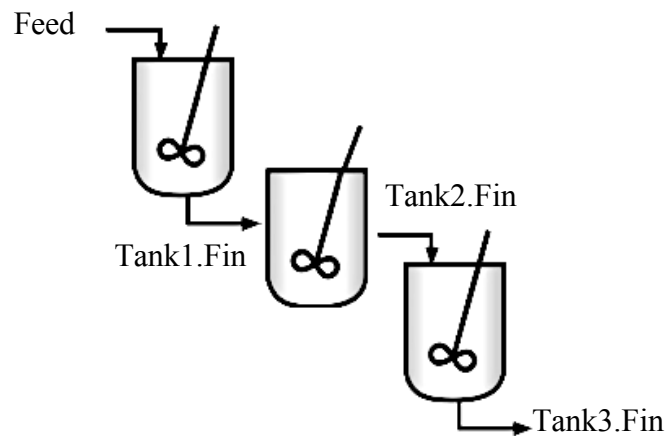


Figura 11. Script ThreeTank, desarrollo del problema de tanques en serie.

Una vez que se ha hecho referencia a la base de datos que se utilizará, se introducen los datos del problema. Primeramente, se nombra el flowsheet, en este caso ThreeTank, y se especifican las relaciones entre los reactores, además se añaden los datos de entrada, como el caudal total y las alturas iniciales dentro de cada tanque, como se puede observar en la figura anterior.

```

Problems | Console | Palette
Output Level: Normal Output
Simulation of ThreeTank started ...
Number of variables: 7
Number of equations: 6
Number of specifications: 1
Degrees of freedom: 0
Structural differential index: 1
Extra Equations: 4
Extra Variables: 0
Dynamic degrees of freedom: 3
Number of initial Conditions: 3
NLA Solver: C:\Program Files (x86)\EMSO\interface\sundials.dll
Solving the initial condition problem...
NLA solver converged.
DAE Solver: C:\Program Files (x86)\EMSO\interface\dasslc.dll
Integrating the system...
Advancing time from 0 to 10
Advancing time from 10 to 20
Advancing time from 20 to 30
Advancing time from 30 to 40
Advancing time from 40 to 50
Advancing time from 50 to 60
Advancing time from 60 to 70
Advancing time from 70 to 80
Advancing time from 80 to 90
Advancing time from 90 to 100
Simulation of 'ThreeTank' finished successfully in 0.093 seconds.

```

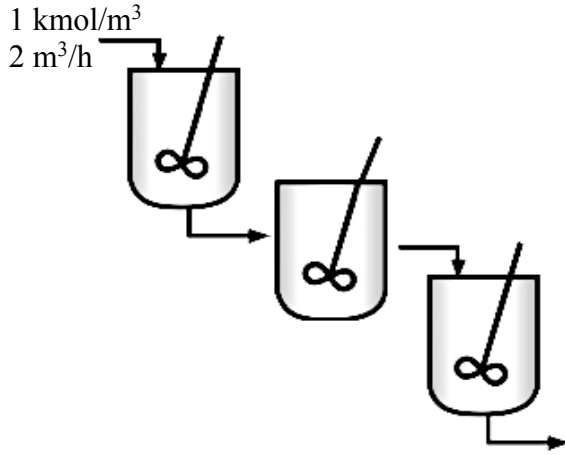
Figura 12. Resultados en la consola.

Al introducir todos los datos del problema, se acciona el botón run y aparecerán todos los resultados en la consola (figura 12). Como se puede observar, es un problema totalmente especificado, el número de ecuaciones y especificaciones coinciden con el número de variables.



### 5.6.2. Optimización de tres reactores continuos de tanque agitado en serie

Tres reactores continuos de tanque agitado se conectan entre sí en serie. El objetivo de este problema es determinar el volumen en cada reactor, sabiendo que el volumen total de todos los reactores no debe sobrepasar los 7 m<sup>3</sup> y minimizando la concentración del componente A en la corriente de salida en el reactor 3. Se conocen los siguientes datos: La concentración molar de entrada es de 1 kmol/m<sup>3</sup>, el caudal volumétrico de entrada son 2 m<sup>3</sup>/h, la constante de velocidad 1 m<sup>1.5</sup>/h/kmol<sup>0.5</sup>.



$$F_o(C_{Ai-1} - C_{Ai}) = r_{Ai}V_i, i \in \{1,2,3\} \quad (4)$$

$$r_{Ai} = k C_{Ai}^{1.5}, i \in \{1,2,3\} \quad (5)$$

$$\left( \begin{array}{l} \min C_{A3}(V) \\ s.t. \\ Eq. (4), Eq. (5) \\ \sum_{i=1}^3 V_i - 7.0 \leq 0 \\ V = [V_1, V_2, V_3]^T \\ V \in \mathfrak{R}_+^3 \end{array} \right) \quad (6)$$

Donde  $C_{Ai}$  es la concentración molar en kmol/m<sup>3</sup> de cada reactor  $i$ ,  $V_i$  el volumen en m<sup>3</sup> de cada reactor,  $F_o$  el caudal de entrada en m<sup>3</sup>/h,  $k$  la constante de velocidad en m<sup>1.5</sup>/kmol<sup>0.5</sup> h y  $r_{Ai}$  la velocidad de reacción en kmol/m<sup>3</sup> h de cada reactor.

En este problema, a diferencia del problema anterior, se especifican todas las variables, las ecuaciones de balance de materia (4) y las ecuaciones de velocidad (5). Por lo tanto, no se hace uso de la base de datos de EMSO.

```

flow_vol    as Real(Unit = 'm^3/h');
conc_mass   as Real(Unit = 'kmol/m^3', Lower = -1e-6);
volum       as Real(Unit = 'm^3');
mass        as Real(Unit = 'kmol');

Model Stream
  VARIABLES
    Fo as flow_vol (Brief = "volumetric flow rate");
    Ca as conc_mass;
end

Model Source
  VARIABLES
    out Outlet as Stream;
end

Model Sink
  VARIABLES
    in Inlet as Stream;
end

```

Figura 13. Script stream\_model.

En primer lugar, en el script stream\_model (figura 13) se nombran todos los variables que intervienen en las corrientes, especificando sus unidades físicas.

```

using "stream_model";

Model Reactor
  VARIABLES
    in Inlet as Stream;
    out Outlet as Stream;

    v as volum (Brief = "Reactor volum");
    r as Real (Brief = "Reactor rate of component A", Unit = 'kmol/m^3/h');

  EQUATIONS
    "Molar balance"
    Inlet.Fo * Inlet.Ca - Outlet.Fo * Outlet.Ca - r * v = 0 * 'kmol/h';

    "Global balance"
    Inlet.Fo = Outlet.Fo;
end

```

Figura 14. Script reactor\_model.

En el reactor\_model se añaden los balances de materia, ecuación (4), utilizando el comando using para llamar al stream\_model y correlacionar los scripts.

```

using "reactor_model";

FlowSheet CSTR
PARAMETERS
k as Real(Brief = "Reaction constant", Unit = '(m^1.5/kmol^0.5)/h');

VARIABLES
Ca_out as conc_mass;

DEVICES
Reactor1 as Reactor;
Reactor2 as Reactor;
Reactor3 as Reactor;
Feed as Source;
Outlet_R4 as Sink;

CONNECTIONS
Feed.Outlet to Reactor1.Inlet;
Reactor1.Outlet to Reactor2.Inlet;
Reactor2.Outlet to Reactor3.Inlet;
Reactor3.Outlet to Outlet_R4.Inlet;

SPECIFY
Feed.Outlet.Fo = 2 * 'm^3/h';
Feed.Outlet.Ca = 1 * 'kmol/m^3';
Reactor1.v = 2 * 'm^3';
Reactor2.v = 2 * 'm^3';
Reactor3.v = 2 * 'm^3';

SET
k = 1 * '(m^1.5/kmol^0.5)/h';

EQUATIONS
Reactor1.r = k * Reactor1.Outlet.Ca^1.5;
Reactor2.r = k * Reactor2.Outlet.Ca^1.5;
Reactor3.r = k * Reactor3.Outlet.Ca^1.5;
Ca_out = Reactor3.Outlet.Ca;

OPTIONS
TimeStep = 0.1;
TimeEnd = 2;
TimeUnit = 'h';

end

```

Figura 15. Script cstr.

En la figura 15, se puede observar el archivo script «cstr» que describe el diagrama de flujo, definiendo todos los equipos que utiliza (sección «DEVICES»), las conexiones entre los distintos equipos (sección «CONNECTIONS»), la ley cinética en cada reactor, ecuación (5) (sección «EQUATIONS»), y las especificaciones del problema (sección «SPECIFY»). Si sólo se pretende resolver el problema sin optimizar se fijan los valores de los volúmenes de cada reactor, en este caso se ha optado por un valor de 2 m<sup>3</sup> para cada reactor, y simplemente se acciona el botón run en el script cstr.

```

using "cstr";

Optimization NLP_CSTR as CSTR
  MINIMIZE
    Reactor3.Outlet.Ca;

  FREE
    Reactor1.v;
    Reactor2.v;
    Reactor3.v;

  EQUATIONS
    Reactor1.v + Reactor2.v + Reactor3.v <= 7 * 'm^3';
    Reactor1.v > 0 * 'm^3';
    Reactor2.v > 0 * 'm^3';
    Reactor3.v > 0 * 'm^3';

  OPTIONS
    Dynamic = false;
    NLPsolveNLA = true;
    NLPsolver (File = "ipopt_emso", Hessian_approximation = "exact");
    #Hessian_approximation = "limited-memory"

end

```

$$\begin{aligned}
 & \min C_{A3}(V) \\
 & s.t. \\
 & \quad Eq. (4), Eq. (5) \\
 & \quad \sum_{i=1}^3 V_i - 7.0 \leq 0 \\
 & \quad V = [V_1, V_2, V_3]^T \\
 & \quad V \in \mathbb{R}_+^3
 \end{aligned}$$

Figura 16. Script NLP\_cstr, optimización NLP.

Para optimizar el problema tal formulado en (6), se utiliza el script «NLP\_cstr» haciendo referencia al anterior script «cstr», se añaden las restricciones (sección «EQUATIONS»), la variable a minimizar (sección «MINIMIZE»), y se resuelve mediante el comando NLPsolver (sección «OPTIONS»).

### 5.6.3. Planificación de la producción

La producción del componente C tiene lugar gracias a una serie de procesos, como se puede observar en la figura 17. El componente C se fabrica a partir del componente B, que puede producirse mediante el proceso 1 o puede comprarse a un proveedor. Para dar lugar al producto se puede elegir entre dos procesos, el proceso 2 con un rendimiento del 82% o el proceso 3 con un rendimiento del 95%. Una vez se tiene el producto C, según las toneladas/horas que se obtenga se puede vender a un precio u a otro.

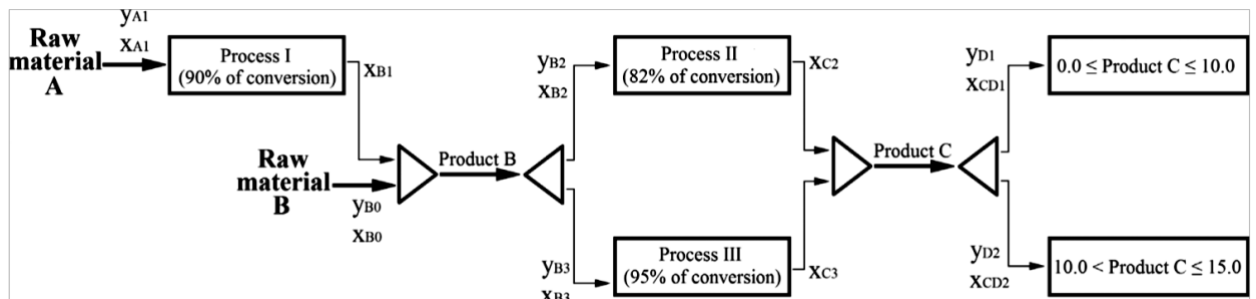


Figura 17. Diagrama de producción del componente C.

En la siguiente tabla se puede observar un resumen de los gastos y precios de venta del proceso de producción.

**Tabla 1.** Gastos de producción y precios de venta.

Proceso	Costes de capital (\$/Ton)	Costes de operación (\$/Ton)
I	1000	250
II	1500	400
III	2000	550

Materia prima	Costes (\$/Ton)
A	500
B	950

Componente C	Precio venta (\$/Ton)
$x_{CD} \leq 10$	1800
$10 < x_{CD} \leq 15$	1500

Se plantean las ecuaciones con las variables binarias, en este caso y, y las variables x que representan las toneladas/hora. Con las siguientes restricciones:  $x_{A1}$  tiene que ser como máximo 16 toneladas/hora, la producción de  $x_C$  no debe sobrepasar las 15 toneladas/hora, además de la opción entre un proceso u otro. Con toda esta información se plantean las siguientes ecuaciones lineales mixtas que se conocen como “Mixed Integer Linear Problems (MILP)”.

A continuación, se representa un ejemplo detallado de las disyunciones, ecuaciones, restricciones y variables binarias utilizadas.

Disyunciones para cada proceso

$$\left[ \begin{array}{c} Y_{A1} \\ X_{B1} = 0.9 X_{A1} \\ C_{1A} = 1000 + 750 X_{A1} \\ X_{A1} \leq 16 \end{array} \right] \vee \left[ \begin{array}{c} Y_{B0} \\ C_{1B} = 950 X_{B0} \\ X_{B0} \leq 50 \end{array} \right]$$

$$\left[ \begin{array}{c} Y_{B2} \\ X_{C2} = 0.82 X_{B2} \\ C_2 = 1500 + 400 X_{B2} \\ X_{B2} \leq 30 \end{array} \right] \vee \left[ \begin{array}{c} Y_{B3} \\ X_{C3} = 0.95 X_{B3} \\ C_3 = 2000 + 550 X_{B3} \\ X_{B3} \leq 30 \end{array} \right]$$

$$\left[ \begin{array}{c} Y_{D1} \\ P_{D1} = 1800 X_{CD1} + 18000 \\ X_{CD1} \leq 10 \end{array} \right] \vee \left[ \begin{array}{c} Y_{D2} \\ P_{D2} = 1500 X_{CD2} - 15000 \\ 10 \leq X_{CD2} \leq 15 \end{array} \right]$$

Donde X (Ton/h) son las toneladas producidas por hora e Y las variables binarias de cada proceso:

$Y_{A1}$ : Proceso I

$Y_{B0}$ : Compra del componente B

$Y_{B2}$ : Proceso II

$Y_{B3}$ : Proceso III

$Y_{D1}$ : Toneladas producidas inferiores a 10 Ton/h

$Y_{D2}$ : Toneladas producidas superiores a 10 Ton/h

C (\$/h) el coste total de cada proceso por hora y P (\$/h) el beneficio obtenido por hora de las ventas:

$C_{1A}$ : Coste total del proceso I

$C_{1B}$ : Coste total de la compra del componente B

$C_2$ : Coste total del proceso II

$C_3$ : Coste total del proceso III

$P_{D1}$ : Beneficio total por toneladas producidas inferiores a 10 Ton/h

$P_{D2}$ : Beneficio total por toneladas producidas superiores a 10 Ton/h

Variables binarias

$$[Y_{A1}] \quad \underline{\vee} \quad [Y_{B0}]$$

$$[Y_{B2}] \quad \underline{\vee} \quad [Y_{B3}]$$

$$[Y_{D1}] \quad \underline{\vee} \quad [Y_{D2}]$$

Función objetivo

$$Profit = P_{D1} + P_{D2} - [C_{1A} + C_{1B} + C_2 + C_3]$$

Disyunciones

$$Y_{A1} + Y_{B0} \leq 2$$

$$Y_{B2} + Y_{B3} = 1$$

$$Y_{D1} + Y_{D2} = 1$$

Restricciones

$$X_{A1} \leq 16 Y_{A1} \quad X_{B0} \leq 50 Y_{B0}$$

$$X_{B2} \leq 30 Y_{B2} \quad X_{B3} \leq 30 Y_{B3}$$

$$X_{CD1} \leq 10 Y_{D1} \quad 10 Y_{D2} \leq X_{CD2} \quad X_{CD2} \leq 15 Y_{D2}$$

$$X_{C2} - X_{C3} \leq 15$$

Balances de materia

$$X_{B2} + X_{B3} = X_{B1} + X_{B0}$$

$$X_{CD1} + X_{CD2} = X_{C2} + X_{C3}$$

Resumen de las ecuaciones y restricciones utilizadas

$$\left( \begin{array}{l} \max_{x,y} P = 100[18x_{CD1} + 180y_{D2} + 15(x_{CD2} - 10y_{D2}) - 10y_{A1} - 7.5x_{A1} - 9.5x_{B0} - 15y_{B2} \\ \quad - 4x_{B2} - 20y_{B3} - 5.5x_{B3}] \\ \\ s.t. \\ \begin{array}{lll} x_{B1} - 0.90x_{A1} = 0; & y_{B2} + y_{B3} - 1 = 0; & x_{B2} + x_{B3} - x_{B1} - x_{B0} = 0; \\ x_{C2} - 0.82x_{B2} = 0; & x_{C3} - 0.95x_{B3} = 0; & x_{CD1} + x_{CD2} - x_{C2} - x_{C3} = 0; \\ y_{D1} + y_{D2} - 1 = 0; & x_{A1} - 16y_{A1} \leq 0; & x_{B0} - 50y_{B0} \leq 0; \\ y_{A1} + y_{B0} - 2 \leq 0; & x_{B2} - 30y_{B2} \leq 0; & x_{B3} - 30y_{B3} \leq 0; \\ x_{C2} - x_{C3} - 15 \leq 0; & x_{CD1} - 10y_{D1} \leq 0; & x_{CD2} - 15y_{D2} \leq 0; \quad 10y_{D2} - x_{CD2} \leq 0 \end{array} \\ \\ \mathbf{x} = [x_{A1}, x_{B0}, x_{B1}, x_{B2}, x_{B3}, x_{C2}, x_{C3}, x_{CD1}, x_{CD2}]^T \\ \mathbf{y} = [y_{A1}, y_{B0}, y_{B2}, y_{B3}, y_{D1}, y_{D2}]^T \\ \mathbf{y} \in \{0, 1\}^6 \\ \mathbf{x} \in \mathfrak{R}_+^9 \end{array} \right) \quad (7)$$

En el flowsheet, se plantea el problema fijando las variables de forma que se obtenga una solución, tal como se indica en la figura 18.

```
FlowSheet Production_planning
VARIABLES
yA1 as Integer (Lower = 0, Upper = 1);
yB0 as Integer (Lower = 0, Upper = 1);
yB2 as Integer (Lower = 0, Upper = 1);
yB3 as Integer (Lower = 0, Upper = 1);
yD1 as Integer (Lower = 0, Upper = 1);
yD2 as Integer (Lower = 0, Upper = 1);

xA1 as Real (Lower = 0, Upper = 50);
xB0 as Real (Lower = 0, Upper = 50);
xB1 as Real (Lower = 0, Upper = 50);
xB2 as Real (Lower = 0, Upper = 50);
xB3 as Real (Lower = 0, Upper = 50);
xC2 as Real (Lower = 0, Upper = 50);
xC3 as Real (Lower = 0, Upper = 50);
xCD1 as Real (Lower = 0, Upper = 50);
xCD2 as Real (Lower = 0, Upper = 50);

Profit as Real (Lower = -100000, Upper = 100000);
```

```

EQUATIONS
xB1 - 0.9*xA1 = 0;
yB2 + yB3 - 1 = 0;
xB2 + xB3 - xB1 - xB0 = 0;
xC2 - 0.82*xB2 = 0;
xC3 - 0.95*xB3 = 0;
xCD1 + xCD2 - xC2 - xC3 = 0;
yD1 + yD2 - 1 = 0;

Profit = 1800*xCd1 + 18000*yD2 + (xCd2 - 10*yD2)*1500 - 1000*yA1
- 750*xA1 - 950*xB0 - 1500*yB2 - 400*xB2 - 2000*yB3 - 550*xB3;

SPECIFY
yA1 = 1;
yB0 = 0;
yB3 = 1;
yD1 = 0;
xB3 = 14.4;
xC2 = 0;
xA1 = 16;
xCD1 = 0;

OPTIONS
Dynamic = false;
NLASolver(RelativeAccuracy = 1e-5);

end

```

Figura 18. Script Production\_planning.

Para saber qué procesos se deben seguir con el objetivo de aumentar el beneficio, se plantea la optimización del proceso indicando la variable a maximizar, en este caso el beneficio (profit), según las ecuaciones (7).

```

using "production_planning.mso";

Optimization Milp_production_planning as Production_planning

MAXIMIZE
Profit;

FREE
yA1;
yB0;
yB3;
yD1;
xB3;
xC2;
xCD1;
xA1;

EQUATIONS
xA1 - 16*yA1 <= 0;
xB0 - 50*yB0 <= 0;
yA1 + yB0 - 2 <= 0;
xB2 - 30*yB2 <= 0;
xB3 - 30*yB3 <= 0;
xC2 + xC3 - 15 <= 0;

xCD1 - 10*yD1 <= 0;
xCD2 - 15*yD2 <= 0;
-xCD2 + 10*yD2 <= 0;

OPTIONS
Dynamic = false;
NLPSolveNLA = false;
NLPSolver(File = "minlp_emso", print_level = 5);

end

```

Figura 19. Script MILP production\_planning.



#### 5.6.4. Optimización de la producción de amoníaco

En este apartado, se realiza una optimización aplicada a la producción de una planta de amoníaco. La corriente de entrada contiene 74% de hidrógeno, 24% de nitrógeno, 1% de argón and 1% of metano (base molar), a 27°C y 10 atm, entra al mezclador 1 con un caudal de 2000 lbmol/h. A continuación, se mezcla con una corriente de recirculación y se comprime a la presión del reactor, 200 atm. El efluente de la compresión se mezcla con una segunda corriente de recirculación que entra en el reactor, teniendo lugar la reacción representada por la ecuación (8) con una conversión del 40%.

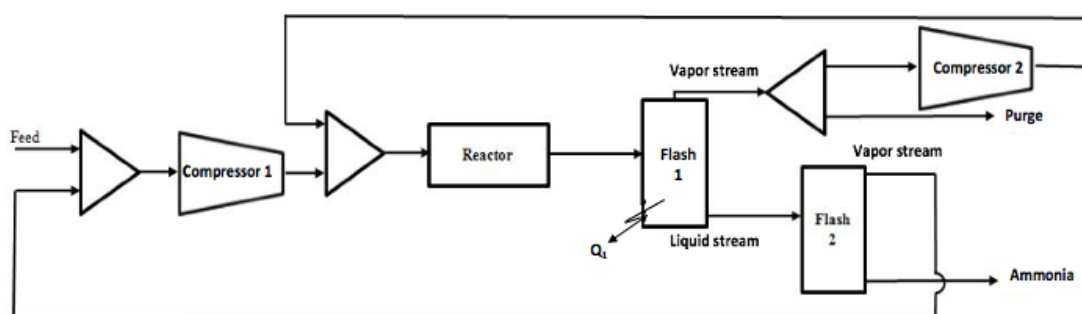


Figura 20. Diagrama de producción de amoníaco.

El caudal de salida del reactor entra en el primer flash donde se separan las corrientes de líquido y vapor. El caudal de salida en fase vapor se divide en dos corrientes, la primera se recircula comprimiendo nuevamente a la presión del reactor y la segunda se elimina por la corriente de purga. El caudal en fase líquido entra en un segundo flash donde se obtiene amoníaco líquido de alta pureza y, por otro lado, el caudal en fase vapor que se recircula al proceso.

Tabla 2. Puntos de ebullición de los distintos componentes que intervienen (Yaws, 1997).

Componente	H <sub>2</sub>	N <sub>2</sub>	Ar	CH <sub>4</sub>	NH <sub>3</sub>
T <sub>B</sub> (°C)	-252,76	-195,8	-185,87	-161,49	-33,43

Es importante destacar que al trabajar con puntos de ebullición tan bajos, como se observa en la tabla 2, no es necesario aportar calor en las unidades flash, por el contrario se elimina calor para facilitar que el producto se obtenga en fase líquida y no gas.

Por otro lado, se utilizan las siguientes ecuaciones para la simulación del proceso en general y para el modelo del reactor.

$$F_{out} H_{out} = F_{in} H_{in} \quad (9)$$

$$F_{out} = F_{in} \left[ 1 - \sum_i x_i^{out} \right] \quad (10)$$

$$x_i^{out} = x_i^{in} + \sum_i (\alpha_i x_i^{in} X_{NH_3}) \quad (11)$$

$$P_{out} = P_{in} \quad (12)$$

Donde i hace referencia a un componente en concreto que interviene en el proceso ( $H_2$ ,  $N_2$ , Ar,  $CH_4$  y  $NH_3$ ).

H (kJ/kmol) es la entalpía de cada corriente.

F (kmol/h) es el caudal molar.

$x_i$  (moles i/moles totales) es la fracción molar de cada componente.

$\alpha_i$  el coeficiente estequiométrico de cada componente.

$X_{NH_3}$  la conversión del reactivo limitante.

P (atm) la presión.

$$Production = Purity \cdot F_{flash\ 2,Liq.} \quad (13)$$

$$Loose = F_{purga} \cdot x_{NH_3}^{purga} \quad (14)$$

$$Purity = x_{NH_3}^{Flash\ 2,Liq.} \quad (15)$$

$$\left( \begin{array}{l} \min \quad |Q_1| + |Q_2| \\ s. t. \quad Eq. (13), \quad Eq. (14), \quad Eq. (15) \\ \\ Loose \leq 0.7 \\ Production \geq 90 \\ \\ F_{purga} \in \mathbb{R}_+^3 \end{array} \right) \quad (16)$$

Donde  $Q_1$  (kW) y  $Q_2$  (kW), son la energía que se elimina en las unidades flash 1 y 2, respectivamente.

$F_{\text{flash 2, Liq}}$  (kmol/h) es el caudal molar en la corriente líquida de salida del flash 2.

La Pureza (Purity) (moles  $\text{NH}_3$ /moles totales) es la fracción molar de amoníaco en la corriente líquida de salida del flash 2.

La Producción (Production) (kmol  $\text{NH}_3$ /h) es el caudal molar de amoníaco en la corriente líquida de salida del flash 2.

Las Perdidas (Loose) (kmol  $\text{NH}_3$ /h) es el caudal molar de amoníaco que se pierde en la corriente de purga.

Para desarrollar el problema en EMSO se ha utilizado la base de datos del programa para el modelo de flash steady (flash stage separator) y el modelo de splitter (separador de corrientes), Scripts que se pueden consultar en el anexo I. Para los demás modelos se especifican todas las ecuaciones y las variables, en especial para el modelo del reactor se especifican las variables y ecuaciones, mediante las ecuaciones (9), (10), (11) y (12).

En el script Ammonia se resuelve el sistema dando valores a las variables, como se observa en la siguiente figura.

```
using "stage_separators/flash";
using "mixers_splitters/splitter";

# A simple ideal compressor
Model Compressor
  PARAMETERS
  outer PP as Plugin(Brief = "External Physical Properties", Type="PP");
  outer NComp as Integer;

  VARIABLES
  in Inlet as stream;
  out Outlet as streamPH;

  EQUATIONS
  "Isentropic expansion"
  PP.VapourEntropy(Outlet.T, Outlet.P, Outlet.z) =
  ... PP.VapourEntropy(Inlet.T, Inlet.P, Inlet.z);

  "Global Molar Balance"
  Inlet.F = Outlet.F;
  "Component Molar Balance"
  Inlet.z = Outlet.z;
end
```

```

# A simple 2 Inlet mixer.
Model Mixer
  PARAMETERS
    outer PP as Plugin(Brief = "External Physical Properties", Type="PP");
    outer NComp as Integer;

    VARIABLES
      in Inlet1 as stream;
      in Inlet2 as stream;
      out Outlet as streamPH;

      EQUATIONS
        "Energy Balance"
        Outlet.F * Outlet.h = Inlet1.F * Inlet1.h + Inlet2.F * Inlet2.h;

        Inlet1.P = Outlet.P;

        "Global Molar Balance"
        Inlet1.F + Inlet2.F = Outlet.F;
        "Component Molar Balance"
        Inlet1.z*Inlet1.F + Inlet2.z*Inlet2.F = Outlet.F * Outlet.z;
      end

Model Reactor
  PARAMETERS
    outer PP as Plugin(Brief = "External Physical Properties", Type="PP");
    outer NComp as Integer;
    NReac as Integer(Default=1);
    stoic(NComp, NReac) as Real (Brief = "Stoichiometric Matrix");
    comp(NReac) as Integer(Default=1, Brief = "Key Component of the reaction");

    VARIABLES
      in Inlet as stream;
      out Outlet as streamPH;
      Outletz(NComp) as fraction;
      X(NReac) as fraction(Brief="Conversion of the key component");

      EQUATIONS
        "Energy Balance"
        Outlet.F * Outlet.h = Inlet.F * Inlet.h;

        "Global Molar Balance"
        Outlet.F = Inlet.F * (1 - sum(Outletz));

        for i in [1:NComp] do
          "Component Molar Balance"
          Outletz(i) = Inlet.z(i) + sum(stoic(i,:)*X*Inlet.z(comp));
        end

        "Normalize the outlet composition"
        Outlet.z * sum(Outletz) = Outletz;

        Outlet.P = Inlet.P;
      end

# Ammonia Process
FlowSheet Ammonia
  PARAMETERS
    PP as Plugin(Brief="Physical Properties", Type="PP",
      Components = ["hydrogen", "nitrogen", "argon", "methane", "ammonia"],
      LiquidModel = "ASRK",
      VapourModel = "ASRK");
    NComp as Integer;
    SET

    NComp = PP.NumberOfComponents;

    DEVICES
      FEED as source;
      C101 as Compressor;
      R101 as Reactor;
      F101 as flash_steady;
      F102 as flash_steady;
      S101 as splitter2;
      M101 as Mixer;
      M102 as Mixer;
      C102 as Compressor;

```

```

VARIABLES
purity as fraction(Brief="Purity of the product");
production as flow_mol(DisplayUnit = 'lbmol/h', Brief="Ammonia in the product");
loose as flow_mol(DisplayUnit = 'lbmol/h', Brief="Ammonia in the purge");
Q1 as energy_source;
Q2 as energy_source;

CONNECTIONS
FEED.Outlet      to M101.Inlet1;
M101.Outlet      to C101.Inlet;
C101.Outlet      to M102.Inlet1;
M102.Outlet      to R101.Inlet;
R101.Outlet      to F101.Inlet;
F101.OutletLiquid to F102.Inlet;
F102.OutletVapour to M101.Inlet2;
F101.OutletVapour to S101.Inlet;
S101.Outlet1     to C102.Inlet;
C102.Outlet      to M102.Inlet2;

Q1.OutletQ       to F101.InletQ;
Q2.OutletQ       to F102.InletQ;

SET
R101.comp = 2; # Key component of the reaction
R101.stoic = [-3, -1, 0, 0, 2]; # Stoichiometry of the reaction
# Reaction: 3 H2 + N2 ----> 2 NH3;

SPECIFY
FEED.F = 2000 * 'lbmol/h';
FEED.T = (27 + 273.15) * 'K';
FEED.P = 10 * 'atm';
FEED.Composition = [0.74, 0.24, 0.01, 0.01, 0.0];
#           #      H2,   N2,   Ar,   CH3, NH3;

C101.Outlet.P = 200 * 'atm';
C102.Outlet.P = 200 * 'atm';

R101.X = 0.4; # Conversion of the reactor

F101.OutletVapour.P = 199 * 'atm';
F101.OutletVapour.T = (0 + 273.15) * 'K';

F102.OutletVapour.P = 10 * 'atm';
F102.InletQ = 0 * 'kJ/h';

# We can choose between one of the following specs
S101.FlowRatios(1) = 0.78; # Recycle fraction
#loose = 0.8 * 'lbmol/h'; # Ammonia in the purge

EQUATIONS
production = purity * F102.OutletLiquid.F;
purity = F102.OutletLiquid.z(5);
loose = S101.Outlet2.F * S101.Outlet2.z(5);
#Loose = purge of Ammonia flow rate;

OPTIONS
Dynamic = false;
NLASolver(
    RelativeAccuracy = 1e-5
);
end

```

Figura 21. Script Ammonia.

Como se puede observar en la figura 21, se definen todos los equipos haciendo referencia al modelo que se utiliza (sección «DEVICES»), las variables que no están presentes en los modelos de los equipos, las conexiones entre los distintos equipos (sección «CONNECTIONS»), mediante el comando «SET» se introducen los datos del problema que ya están definidos en los modelos de equipo, pero que no corresponden con esta simulación en concreto. Seguidamente, se introducen las especificaciones del problema (sección «SPECIFY»), y mediante las ecuaciones (13), (14), y (15) se resuelve el problema dando lugar a la simulación de la producción de amoniaco.

Una vez se ha simulado la planta de producción de amoniaco, se pretende optimizar la planta en general, minimizando las pérdidas de amoniaco por la purga, aumentando la

producción de amoniaco, minimizando el intercambio de calor en las unidades flash y manteniendo o mejorando la pureza del producto final.

Para ello se plantea la optimización en EMSO de la simulación Ammonia según las ecuaciones (16), creando un nuevo script que se llama Ammonia\_opt. Es importante hacer referencia a la simulación utilizando el comando “using”.

```
using "ammonia";

Optimization AmmoniaOPT as Ammonia
MINIMIZE

(abs(Q1.OutletQ) + abs(Q2.OutletQ))*1e-6
#loose / 'lbmol/h'
#C102.Pot/'kW'*180
#+ F101.InletQ/'kW'/10
;

FREE
#F101.OutletVapour.T;
S101.FlowRatios(1);

EQUATIONS
#F102.OutletLiquid.z(5) > 0.95;
loose < 0 * 'lbmol/h';
production > 90 * 'lbmol/h';

OPTIONS
Dynamic = false;
NLPsolveNLA = true;
FeasiblePath = true;
# NLPsolver(#File = "complex",
#           #File = "optpp_emso",
#           File = "ipopt_emso",
#           MaxIterations = 300,
#           RelativeAccuracy = 1e-6);
end
```

Figura 22. Script Ammonia\_opt.

$$\begin{aligned} \min \quad & |Q_1| + |Q_2| \\ \text{s.t.} \quad & \text{Production} = \text{Purity} \cdot F_{\text{flash 2,Liq.}} \\ & \text{Loose} = F_{\text{purga}} \cdot x_{\text{NH}_3}^{\text{Purga}} \\ & \text{Purity} = x_{\text{NH}_3}^{\text{Flash 2,Liq.}} \\ & \text{Loose} \leq 0.7 \\ & \text{Production} \geq 90 \\ & F_{\text{purga}} \in \mathbb{R}_+^3 \end{aligned}$$

Los resultados obtenidos se pueden observar en el apartado resultados. Donde se comentan los resultados tanto de la simulación como de la simulación.

#### 5.6.4.1. Análisis económico de la producción de amoniaco

Para hacer una estimación económica del proceso de producción de la planta de amoniaco, se ha seguido el procedimiento detallado en el libro (Turton, Analysis, Synthesis, and Design of Chemical Processes, 2018).

##### a) Coste capital

Primeramente, entre los métodos señalados para el cálculo del coste capital de la planta se ha optado por el método de cálculo modular, ya que es más específico y se tienen en cuenta factores como el tipo de material de construcción, el tipo de equipo utilizado y la presión a la que se trabaja. La ecuación (17) se utiliza para calcular el costo del módulo básico para cada equipo. El costo del módulo básico es la suma de los costos directos e indirectos.

$$C_{BM} = C_p^o F_{BM} \quad (17)$$

Donde  $F_{BM}$  es el factor de costo del módulo desnudo: factor de multiplicación para tener en cuenta los materiales específicos de construcción y la presión de operación.

$C_p^\circ$  costo de compra para las condiciones básicas: equipo fabricado con el material más común, generalmente acero al carbono, y que funciona a presiones cercanas a la ambiental.

Los datos para el costo de compra del equipo, a presión ambiente y utilizando el material de construcción de acero al carbono,  $C_p^\circ$  se ajustan a la siguiente ecuación:

$$\text{Log}_{10} C_p^\circ = K_1 + K_2 \text{Log}_{10}(A) + K_3 [\text{Log}_{10}(A)]^2 \quad (18)$$

Donde A es el parámetro de capacidad o tamaño del equipo. Los datos para  $K_1$ ,  $K_2$  y  $K_3$ , junto con los valores máximo y mínimo usados en la correlación, se encuentran en la tabla A.1. en el apéndice A del libro (Turton, Analysis, Synthesis, and Design of Chemical Processes, 2018). Para calcular las constantes  $K_1$ ,  $K_2$  y  $K_3$ , se han elegido los equipos descritos en la siguiente tabla.

**Tabla 3.** Datos de costos de equipos utilizados en la ecuación (18).

Equipos	$K_1$	$K_2$	$K_3$	Equipo elegido
<b>Compresor</b>	2,2897	1,3604	-0,1027	“Centrifugal compressor”
<b>Reactor</b>	3,5565	0,3776	0,0905	“Horizontal process vessel”
<b>Flash</b>	5,2366	-0,6572	0,35	“Evaporator short tube”

Para el reactor “Horizontal process vessel” se tiene en cuenta el factor  $F_{BM}$  con la siguiente ecuación:

$$F_{BM} = B_1 + B_2 F_M F_P \quad (19)$$

Donde  $B_1$  y  $B_2$  son constantes que dependen del equipo considerado, se encuentran en el libro mencionado anteriormente.  $F_m$  es el factor material que varía según el tipo de material, en este caso es igual a la unidad ya que se trata de acero al carbono.  $F_p$  es el factor de presión, depende de la presión a la que se trabaje y se calcula con la siguiente ecuación:

$$F_{p,vessel} = \frac{\frac{PD}{2[850 - 0.6(P)]} + 0.00315}{0.0063} \quad (20)$$

Donde P es la presión a la que trabaja el reactor en barg y D el diámetro del reactor en metros.

Finalmente, se debe tener en cuenta el efecto del tiempo en el costo total de los equipos. Todos los costes calculados en el procedimiento anterior corresponden al año 2001, para tener un coste más actual se utilizan los índices de coste CEPCI del año 2021 ((FINAL), 2021) mediante la siguiente ecuación.

$$Costes_{2021} = Costes_{2001} \left( \frac{I_{2021}}{I_{2001}} \right) \quad (21)$$

Siendo  $I_{2001} = 394$ ,  $I_{2021} = 637$ .

b) Coste operacional

Hay que tener muy presente los costes de refrigeración en las unidades flash ya que se tiene que bajar de temperatura una media 1000°C. Por lo que es necesario contar con un buen sistema de refrigeración. Además, es importante tener en cuenta los costes de electricidad.

$$Coste_{operacional\ refrigerante} \left( \frac{\$}{año} \right) = Q \text{ (kW)} \cdot Precio\ fluido \left( \frac{\$}{kW\ año} \right) \quad (22)$$

$$Coste_{operacional\ electricidad} \left( \frac{\$}{año} \right) = Coste_{electricidad} \left( \frac{\$}{kW\ año} \right) \cdot W \left( \frac{kW}{año} \right) \quad (23)$$

Siendo Q el calor intercambiado en las unidades flash, y W la potencia de los compresores.

c) Coste total anualizado

Para obtener el coste total, se suma el coste capital y el coste operacional, teniendo en cuenta que el coste capital debe multiplicarse por el factor de anualización, como se observa en la siguiente ecuación.

$$Coste_{total} = Coste_{operacional} + Coste_{capital} \cdot F \quad (24)$$

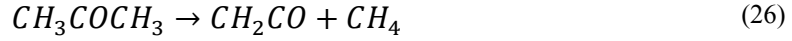
El factor de anulación se calcula con la ecuación (25), siendo i el tipo de interés anual, tomando un valor de 10%, y n el período de amortización, se ha fijado a 10 años.

$$F = \frac{i(1+i)^n}{(1+i)^n - 1} \quad (25)$$



### 5.6.5. Simulación de la producción de anhídrido acético

La producción de anhídrido acético se lleva a cabo en un reactor flujo pistón que se considera adiabático mediante el craqueo de acetona de acuerdo con la siguiente reacción endotérmica:



La corriente de entrada al reactor es de 8000 kg/h con una composición de 98% acetona, 1% keteno y 1% metano, todo en fracciones molares. Se fijan las dimensiones del reactor, un volumen de 1.27 m<sup>3</sup> y una longitud de 2.28 m. La temperatura de entrada al reactor es de 1035 K, la presión 1.6 atmósferas y se considera constante. Por lo tanto, la reacción tiene lugar en fase gaseosa siguiendo una cinética de primer orden con respecto a la acetona. La constante cinética viene dada por la siguiente expresión:

$$k = \left[ e^{34.34 - \left( \frac{34222}{T} \right)} \right] 3600 \quad (27)$$

$$r_j = k C_j \quad (28)$$

Donde la T es la temperatura en Kelvin, k la constante cinética en horas<sup>-1</sup>, C<sub>j</sub> la concentración de cada componente j (acetona, keteno y metano) en kmol/m<sup>3</sup> y r<sub>j</sub> la velocidad de reacción de cada componente j en kmol/m<sup>3</sup> h.

Al tratarse de un reactor flujo pistón se hace uso de las siguientes ecuaciones diferenciales:

Balance molar

$$\frac{dn_j}{dL} = Area \sum_{i=1}^R (r_i \alpha_{ij}) \quad (29)$$

Donde n<sub>j</sub> es el flujo molar del componente j en kmol/h; R el número de reacciones i que tienen lugar en el reactor, en este caso solo una; α<sub>ij</sub> el coeficiente estequiométrico del componente j en la reacción i.

L la longitud del reactor en m; Area el área de la sección transversal del reactor en m<sup>2</sup>.

## Balance de energía

$$\frac{dT}{dL} = \frac{\left(\frac{dQ}{dV}\right) - \sum_{i=1}^R (r_i \Delta H_i)}{\sum_{j=1}^S (n_j C_{pj})} Area \quad (30)$$

$$\Delta H_i = \Delta H_i^* + \int_{T^*}^T \Delta C_{pi} dT \quad (31)$$

$$\Delta C_{pi} = \sum_{j=1}^S (C_{pj} \alpha_{ij}) \quad (32)$$

Donde k el reactivo limitante, en este caso la acetona.

S el número de componentes j.

T la temperatura en el reactor en Kelvin.

C<sub>j</sub> la concentración del componente j en kmol/m<sup>3</sup>.

Q la energía intercambiada, al ser un reactor adiabático en este caso es nulo.

ΔH<sub>i</sub> la entalpía de la reacción i en kJ/kmol.

ΔH<sub>i</sub><sup>\*</sup> la entalpía de la reacción i a la temperatura de referencia T<sup>\*</sup> en kJ/kmol.

T<sup>\*</sup> la temperatura de referencia de la entalpía específica de reacción, en este caso 298K.

C<sub>pj</sub> el calor específico de cada componente j en kJ/kmol K.

Se debe tener en cuenta que a las temperaturas en las que trabaja el reactor todos los componentes están en fase gas. Por lo tanto, es necesario calcular el flujo volumétrico a cada tramo del reactor. Para ello se hace uso de la ecuación de los gases ideales (33) para calcular el flujo volumétrico a la entrada del reactor, suponiendo el gas ideal. Seguidamente se utiliza la ecuación (34) para calcular el flujo volumétrico en cada tramo.

$$PQ_v = nRT \quad (33)$$

$$Q_v = Q_{v.in} \left( \frac{T P_{in}}{T_{in} P} \right) \left[ 1 + \sum_{i=1}^R (\varepsilon_{ik} X_{ik}) \right] \quad (34)$$

$$\varepsilon_{ik} = y_{k0} \frac{\sum_{j=1}^S \alpha_{ij}}{-\alpha_{ik}} \quad (35)$$

Q<sub>v.in</sub> el flujo volumétrico de entrada al reactor en m<sup>3</sup>/h.

Q<sub>v</sub> el flujo volumétrico a lo largo del reactor en m<sup>3</sup>/h.

T<sub>in</sub> la temperatura a la entrada del reactor en Kelvin.

P<sub>in</sub> la presión a la entrada del reactor.

P la presión a lo largo del reactor, que al ser constante P = P<sub>in</sub>.

ε<sub>ik</sub> Factor de expansión en la reacción i.

X<sub>ik</sub> la conversión del reactivo limitante k en la reacción i.

y<sub>k0</sub> la fracción molar a la entrada del reactivo limitante k.

Para desarrollar el problema en EMSO se ha utilizado la base de datos del programa para el modelo del reactor flujo pistón (reactor\_pfr). Como se puede observar en la siguiente figura.

```
using "streams";

Model pfr

ATTRIBUTES
  Pallete      = true;
  Brief        = "Model of a Generic PFR with constant mass holdup";
  Icon         = "icon/pfr";
  Info        =
  "== Requires the information of ==
  * Reaction values
  * Heat of reaction
  * Pressure profile
  ";

PARAMETERS

outer PP          as Plugin          (Brief = "External Physical Properties", Type="PP");
outer NComp       as Integer         (Brief="Number of components");
outer NReac       as Integer         (Brief="Number of reactions");
outer stoic(NComp, NReac) as Real    (Brief = "Stoichiometric Matrix");
outer NDisc       as Integer         (Brief="Number of points of discretization", Default=10);
outer Mw(NComp)   as molweight       (Brief="Component Mol Weight");
outer L           as length          (Brief="Reactor Length");
outer Across      as area            (Brief="Cross section area");

SET

  Mw = PP.MolecularWeight();

VARIABLES

in Inlet as stream (Brief = "Inlet Stream", PosX=0, PosY=0.5076, Symbol="{in}");
out Outlet as stream (Brief = "Outlet Stream", PosX=1, PosY=0.5236, Symbol="{out}");

str(NDisc+1) as vapour_stream;
vol(NDisc+1) as vol_mol;
rho(NDisc+1) as dens_mass;

q(NDisc) as heat_rate;

M(NComp, NDisc) as mol (Brief = "Molar holdup");
Mt(NDisc) as mol (Brief = "Molar holdup");
C(NComp, NDisc) as conc_mol (Brief="Components concentration", Lower=-1e-6);
E(NDisc) as energy (Brief="Total Energy Holdup on element");
r(NReac, NDisc) as reaction_mol;
Hr(NReac, NDisc) as heat_reaction;
#Hf(NComp, NDisc) as heat_reaction;

EQUATIONS

"Inlet boundary"
str(1).F = Inlet.F;
str(1).T = Inlet.T;
str(1).P = Inlet.P;
str(1).z = Inlet.z;

"Outlet boundary"
Outlet.F = str(NDisc+1).F;
Outlet.T = str(NDisc+1).T;
Outlet.P = str(NDisc+1).P;
Outlet.z = str(NDisc+1).z;
Outlet.h = str(NDisc+1).h;
Outlet.v = str(NDisc+1).v;

for z in [1:NDisc] do
  for c in [1:NComp] do
    "Component Molar Balance"
    diff(M(c,z)) = (str(z).F*str(z).z(c) - str(z+1).F*str(z+1).z(c))
    + sum(stoic(c,:)*r(:, z)) * Across*L/NDisc;
  end

  "Energy Balance"
  diff(E(z)) = str(z).F*str(z).h - str(z+1).F*str(z+1).h +
  | sum(Hr(:,z)*r(:,z)) * Across*L/NDisc - q(z);

  "Energy Holdup"
  E(z) = Mt(z)*str(z+1).h - str(z+1).P*Across*L/NDisc;
end
```

```

"mass flow is considered constant"
str(z+1).F*vol(z+1)*rho(z+1) = str(z).F*vol(z)*rho(z);

"Molar concentration"
C(:,z) * Across*L/NDisc = M(:,z);

"Geometrical constraint"
Across*L/NDisc = Mt(z) * vol(z);

"Molar fraction"
str(z+1).z * Mt(z) = M(:,z);

#Hf(:,z) = PP.IdealGasEnthalpyOfFormation(str(z+1).T);
#Hr(:,z) = -sum(stoic*Hf(:, z));
end

for z in [1:NDisc+1] do
"Specific Volume"
vol(z) = PP.VapourVolume(str(z).T, str(z).P, str(z).z);

"Specific Mass"
rho(z) = PP.VapourDensity(str(z).T, str(z).P, str(z).z);
end
end

```

Figura 23. Script del modelo PFR (base de datos EMSO).

El modelo del reactor flujo pistón presente en EMSO requiere que se especifique la cinética de la reacción, el calor de reacción y el perfil de presión dentro del reactor. Datos que se especifican en el flowsheet de la producción de anhídrido acético.

```

using "reactors/pfr";
FlowSheet PFR_AceticAnhydride
PARAMETERS
    PP as Plugin (Brief="Physical Properties",
        Type = "PP",
        Components = ["acetone", "ketene", "methane" ],
        LiquidModel = "PR",
        VapourModel = "PR",
    );
    NComp as Integer;
SET
    NComp = PP.NumberOfComponents;
DEVICES
    s1 as source;
    Reac as pfr;
CONNECTIONS
    s1.Outlet to Reac.Inlet;
SET
    Reac.NDisc = 15; #Number of points of discretization
    Reac.L = 2.28 * 'm';
    Reac.Across = 1.27*'m^3'/Reac.L;
    # Only one reaction
    Reac.NReac = 1;
    # Reaction 1: Acetone -> Ketene + Methane
    Reac.stoic(:,1) = [-1, 1, 1];
SPECIFY
    # Feed specification
    s1.Fw = 8000 * 'kg/h';
    s1.T = 1035 * 'K';
    s1.P = 1.6 * 'atm';
    s1.Composition = [0.98, 0.01, 0.01]; #Stream Molar Fraction
    # Adiabatic
    Reac.q = 0 * 'J/s';

```

```

EQUATIONS
for z in [1:Reac.NDisc] do
  "Reaction Rate = k*C(1)"
  Reac.r(1,z) = exp(34.34 - (34222 * 'K')...
    / Reac.str(z).T)*'1/s' * Reac.C(1,z);
  "Heat of reaction"
  Reac.Hr(1,z) = -80.77 * 'kJ/mol';
  "Pressure Drop (no pressure drop)"
  Reac.str(z+1).P = Reac.str(z).P;
end
INITIAL
for z in [2:Reac.NDisc+1] do
  Reac.str(z).T = Reac.Inlet.T;
  Reac.str(z).z(1:NComp) = Reac.Inlet.z(1:NComp);
end
end

```

*Figura 24. Script Flowsheet AceticAnhydride.*

Es importante resaltar que para resolver las ecuaciones diferenciales del reactor flujo pistón EMSO utiliza un método de discretización, el método de Euler de primer orden. Por lo tanto, divide el reactor en pequeños tramos que se resuelven consecutivamente. La variable NDisc indica el número de tramos en el que se divide el reactor.

## 6. Resultados

### 6.1. Resultados de los casos de estudio

#### 6.1.1. Simulación de tres tanques en serie

Para representar los resultados se añade una gráfica con el botón plot y se introducen las variables a representar. Una vez representadas las gráficas se obtienen los siguientes resultados, donde se puede observar como varían los caudales de entrada (Fin), de salida (Fout) y el nivel de fluido (Level) en cada tanque con respecto al tiempo en segundos.

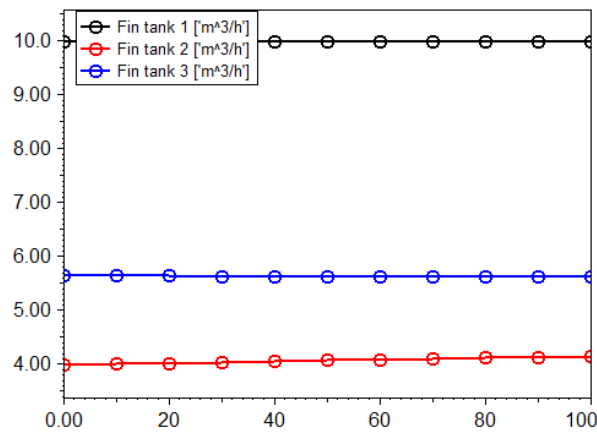


Figura 25. Caudales de entrada frente al tiempo (s).

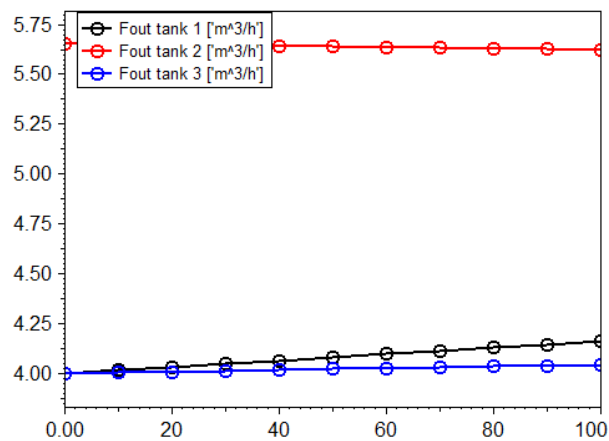


Figura 26. Caudales de salida frente al tiempo (s).

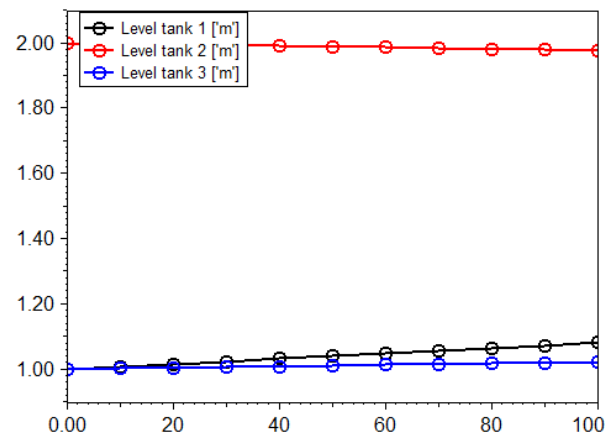


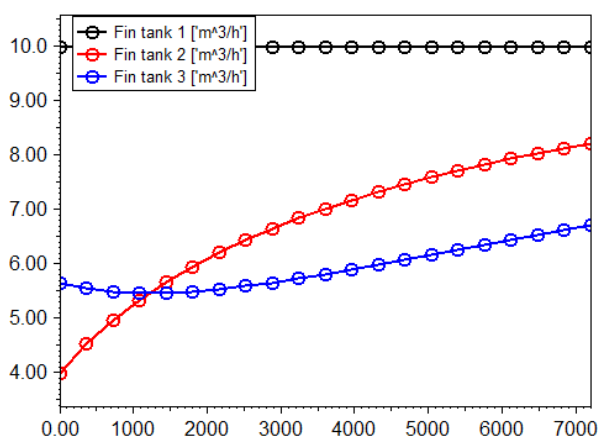
Figura 27. Altura de fluido frente al tiempo (s).

Como se puede observar en las figuras anteriores, hay muy poca variación en las gráficas, esto es debido a que no se ha especificado el intervalo de tiempo dentro del problema.

Para poder observar mejor la tendencia de cada gráfica se especifica el intervalo de tiempo en el que se quiere representar, para ello se añade el comando OPTIONS antes de END en el flowsheet ThreeTank. Tal como se indica a continuación.

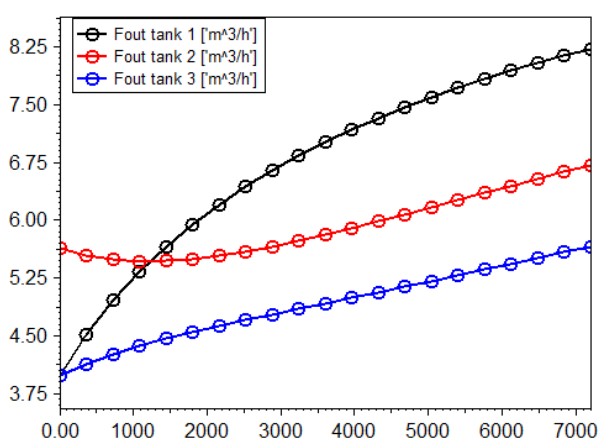
```
OPTIONS
TimeStep = 0.1;
TimeEnd = 2;
TimeUnit = 'h';
```

Obteniendo los siguientes resultados.



**Figura 28.** Caudal de entrada frente al tiempo (s).

Como se puede observar en la figura 28, el caudal de entrada del reactor 1 es constante, como era de esperar ya que previamente se ha fijado para que no varíe, mientras los caudales de los reactores 2 y 3 aumentan ligeramente, destacando el caudal de entrada en el reactor 2 que aumenta exponencialmente con el tiempo.



**Figura 29.** Caudal de salida frente al tiempo (s).

Los caudales de salida aumentan en general, el reactor 1 aumenta exponencialmente, mientras los reactores 2 y 3 tienen un crecimiento bastante paralelo, estando el caudal del reactor 2 por encima del reactor 3.

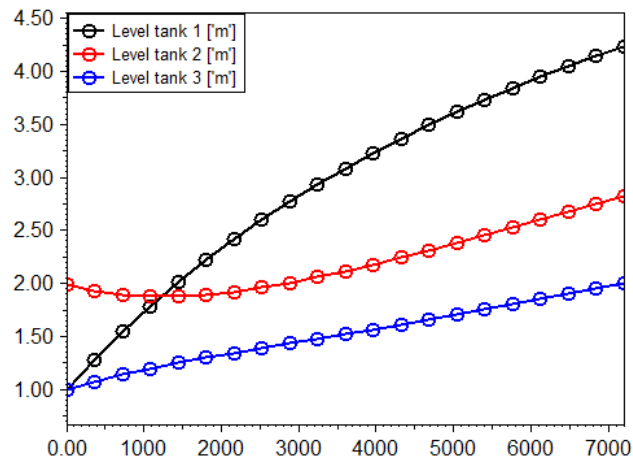


Figura 30. Nivel dentro del reactor frente al tiempo (s).

En la figura 30 se puede observar como el nivel dentro del tanque aumenta en general sin sobrepasar 4.5 m, estando muy por encima el nivel del reactor 1.

Debido a que se utiliza la base de datos del programa, las variables que influyen en las ecuaciones de los reactores vienen ya especificadas, pero si se quiere modificar algún valor sin tener que añadir todas las variables termodinámicas, se utiliza el comando SET, y se especifican las variables que se desee cambiar.

```
FlowSheet ThreeTank
VARIABLES
Feed as flow_vol;

DEVICES
Tank1 as tank_simplified;
Tank2 as tank_simplified;
Tank3 as tank_simplified;

CONNECTIONS
Feed to Tank1.Fin;
Tank1.Fout to Tank2.Fin;
Tank2.Fout to Tank3.Fin;

SPECIFY
Feed = 10 * 'm^3/h';

INITIAL
Tank1.Level = 1 * 'm';
Tank2.Level = 2 * 'm';
Tank3.Level = 1 * 'm';

SET
Tank2.k = 8 * 'm^2.5/h';
Tank2.A = 4 * 'm^2';

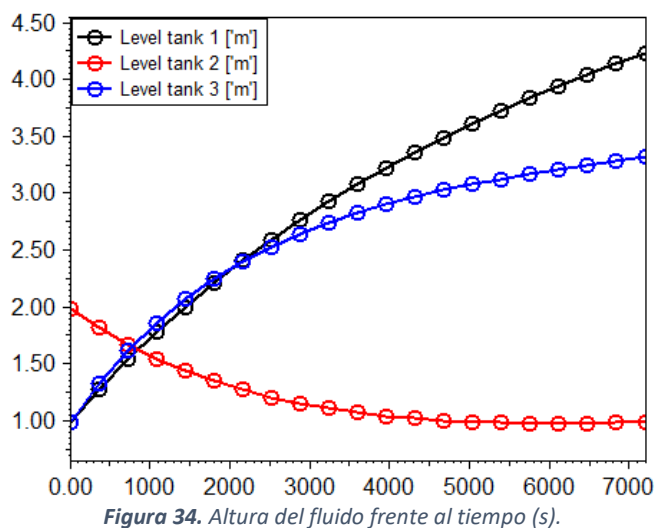
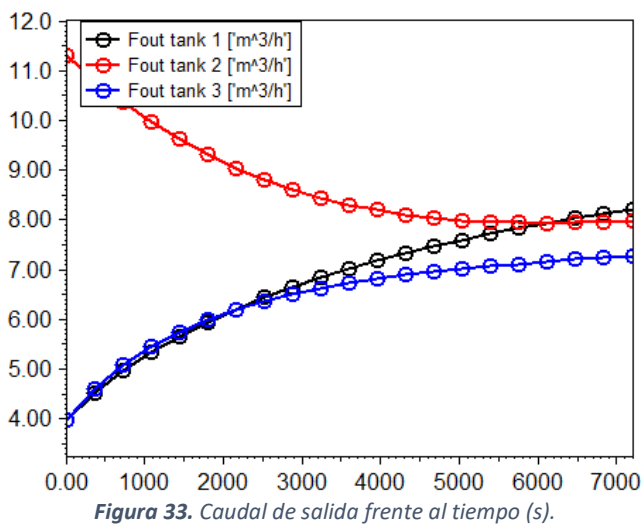
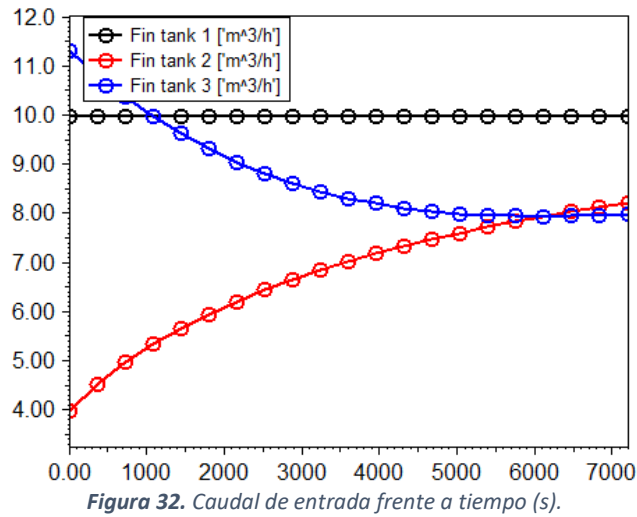
OPTIONS
TimeStep = 0.1;
TimeEnd = 2;
TimeUnit = 'h';

end
```

Figura 31. Script ThreeTank con el comando SET.

En este caso se han cambiado la constante k y el área, en el reactor 2. Finalmente, se obtienen los siguientes resultados.





Como se puede observar en las figuras anteriores al aumentar la constante de velocidad y el área en el reactor 2 (se sabe que los valores aumentan ya que se comparan con los valores fijados en la base de datos del programa presentes en el script tank\_simplified, aumenta el caudal de entrada en el reactor 3 y descende el nivel en el reactor 2, aumentando considerablemente el nivel en el reactor 3.

### 6.1.2. Optimización de tres reactores continuos de tanque agitado en serie

Una vez se han fijado los volúmenes de los reactores, todos igual  $2 \text{ m}^3$ , se especifican las variables de concentración molar a la entrada, el caudal volumétrico de entrada, y la constante de velocidad. Primeramente, se realiza la simulación y posteriormente, mediante las restricciones nombradas en el apartado anterior, se realiza la optimización. Finalmente se obtienen los siguientes resultados de los tres reactores continuos de tanque agitado en serie.

**Tabla 4.** Resultados Cstrs en serie.

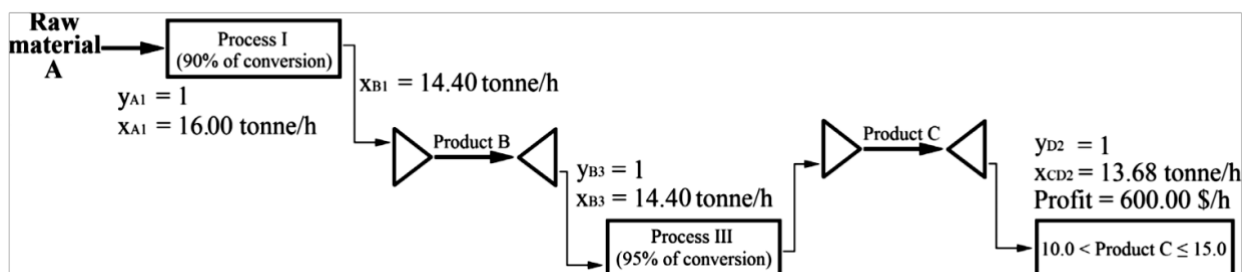
Reactor	Simulación		Optimización	
	V ( $\text{m}^3$ )	Ca ( $\text{kmol}/\text{m}^3$ )	V ( $\text{m}^3$ )	Ca ( $\text{kmol}/\text{m}^3$ )
1	2	0,569840	2,07782	0,562136
2	2	0,356755	2,33159	0,335543
3	2	0,239527	2,59059	0,210472

	Simulación	Optimización
$V_{\text{total}} (\text{m}^3)$	6,00	7,00

En la tabla 4 se pueden observar los resultados de la simulación al lado izquierdo y los resultados de la optimización al lado derecho, ambos resultados exponen las concentraciones y volúmenes de cada reactor. Se puede observar que hay un aumento considerable de los volúmenes de los reactores, con respecto a la simulación. Además, hay una disminución de las concentraciones en general, lo que es de esperarse ya que si aumentan los volúmenes de los reactores las sustancias estarán más diluidas. Como resultado, se consigue minimizar la concentración de salida en el reactor 3, además se cumple la restricción de no sobrepasar los  $7 \text{ m}^3$  de volumen total.

### 6.1.3. Planificación de la producción

Aunque se hallan fijado previamente algunas de las variables para obtener un resultado, mediante la optimización estas variables cambian de valor conforme al objetivo buscado.



**Figura 35.** Resultado de la optimización del Production planning.

Finalmente, es mejor comprar el material A y producir B por medio del proceso I que comprar el producto B directamente. A continuación, es mejor obtener el producto C mediante el proceso III y vender entre 10 y 15 toneladas a la hora. De esta manera, se obtiene un beneficio de 600\$ a la hora.

#### 6.1.4. Optimización de la producción de amoniaco

Se resuelve el problema mediante el solver NLA (non-linear algebraic), obteniendo los siguientes resultados.

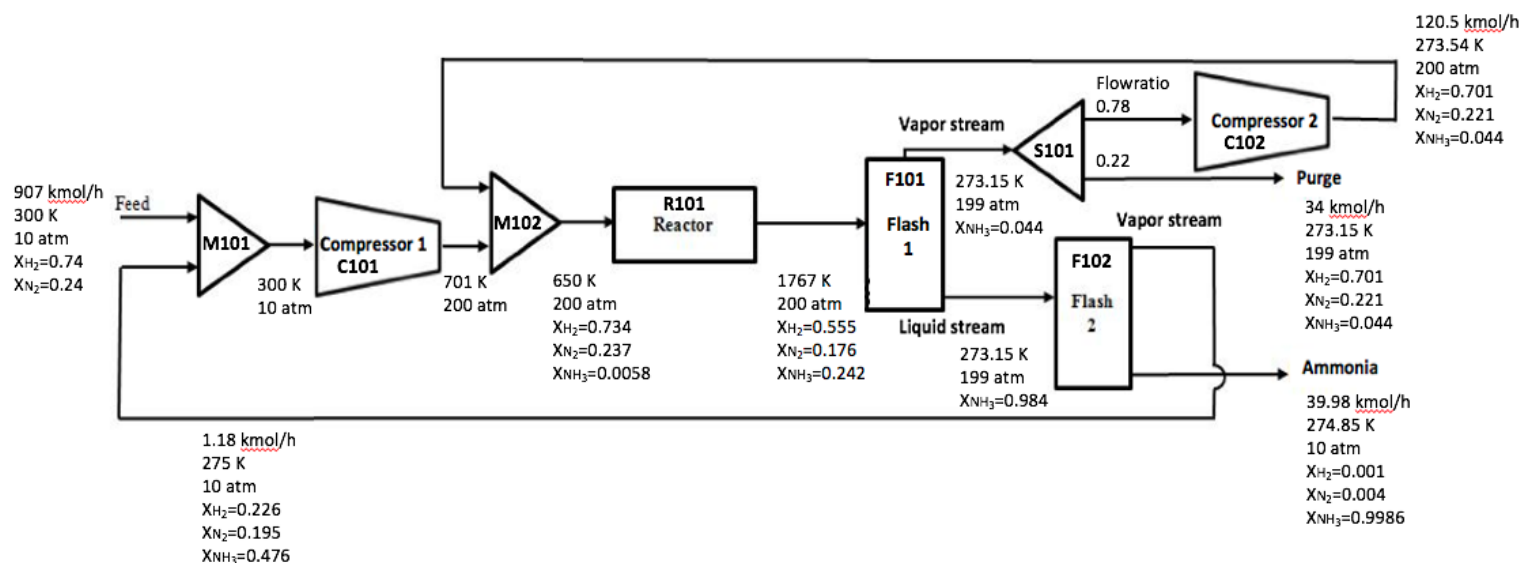


Figura 36. Resultados de la simulación Ammonia.

En la siguiente tabla se pueden observar los resultados de la simulación, se exponen los datos obtenidos de la pureza del amoniaco, el caudal y las pérdidas en la corriente de purga.

Tabla 5. Resultados de la simulación Ammonia.

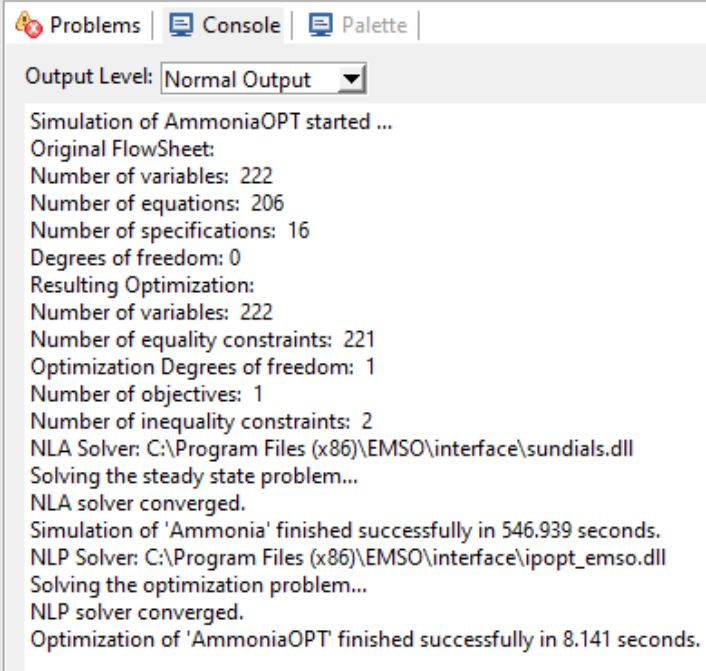
Resultados	Pureza (%)	Producción amoniaco (kmol/h)	Pérdida de amoniaco (kmol/h)
Simulación	99,86%	39,93	1,51

Para encontrar el resultado óptimo se ha llevado a cabo un estudio de la optimización en EMSO, variando las restricciones tanto en la producción mínima de amoniaco como en las perdidas máximas de amoniaco. En la siguiente tabla se observan los distintos resultados de la optimización.

Tabla 6. Resultados de la optimización Ammonia en EMSO, variando las restricciones.

Production (kmol/h)	Loose (kmol/h)	Pureza (%)	Producción NH <sub>3</sub> (kmol/h)	Pérdida NH <sub>3</sub> (kmol/h)
Production > 45	Loose < 0,5	99,86%	45,36	0,403
Production > 25	Loose < 0,35	99,86%	45,44	0,322
Production > 35	Loose < 0,2	99,86%	45,59	0,186
Production > 45	Loose < 0,0	99,86%	45,78	0,000

Como se puede observar en la tabla 6 conforme disminuyen las pérdidas de amoniaco por medio de la corriente de purga, aumenta la producción sin apenas variaciones en la pureza. Gracias a la optimización realizada en EMSO la planta de producción de amoniaco ha mejorado considerablemente, ya que se ha logrado aumentar la producción de amoniaco, eliminar completamente las pérdidas de amoniaco por la corriente de purga, y todo ello manteniendo una alta pureza en el amoniaco producido.



```

Problems | Console | Palette
Output Level: Normal Output
Simulation of AmmoniaOPT started ...
Original FlowSheet:
Number of variables: 222
Number of equations: 206
Number of specifications: 16
Degrees of freedom: 0
Resulting Optimization:
Number of variables: 222
Number of equality constraints: 221
Optimization Degrees of freedom: 1
Number of objectives: 1
Number of inequality constraints: 2
NLA Solver: C:\Program Files (x86)\EMSO\interface\sundials.dll
Solving the steady state problem...
NLA solver converged.
Simulation of 'Ammonia' finished successfully in 546.939 seconds.
NLP Solver: C:\Program Files (x86)\EMSO\interface\ipopt_emso.dll
Solving the optimization problem...
NLP solver converged.
Optimization of 'AmmoniaOPT' finished successfully in 8.141 seconds.

```

*Figura 37. Resultados de la consola en la optimización de amoniaco.*

Por otro lado, el tiempo que tarda el programa EMSO en optimizar la producción de amoniaco son 8,141 segundos, como se puede observar en la figura 37. Teniendo en cuenta la gran cantidad de ecuaciones que intervienen en el proceso, cabe resaltar que el tiempo indicado en la figura 37 sólo hace referencia al tiempo que tarda el programa en optimizar, una vez se ha simulado previamente el proceso. Es decir, no incluye el tiempo que tarda el programa en simular la producción de amoniaco, pero este tiempo son 2,391 segundos. El tiempo total de simulación y optimización del proceso son 10,532 segundos.

#### 6.1.4.1. Resultados del análisis económico

Una vez calculada la estimación económica necesaria para llevar a cabo la producción, mediante el procedimiento descrito en el apartado análisis económico, se obtienen los siguientes resultados. Teniendo en cuenta que se estima una amortización a 10 años del capital invertido.

*Tabla 7. Coste capital de los equipos en la producción amoniaco (amortización a 10 años).*

<b>Equipos</b>	<b>C<sub>p</sub><sup>o</sup></b>	<b>F<sub>BM</sub></b>	<b>C<sub>BM</sub> 2021 (\$/año)</b>
<b>2 Compressor</b>	3527	2,8	3194
<b>2 flash</b>	35620	2,9	83160
<b>Reactor</b>	138070	41,7	540963

En la tabla 7 se puede observar una estimación aproximada de los costes de los diferentes equipos que intervienen en el proceso: 2 Compresores, 2 unidades flash y el coste estimado del reactor. La mayor parte del presupuesto lo abarca el reactor, seguido de las unidades flash y los compresores.

*Tabla 8. Resultados del análisis económico.*

<b>Coste operacional (\$/año)</b>	<b>Coste capital (\$/año)</b>	<b>Coste total (\$/año)</b>
892956	627317	995049

Por otro lado, hay que tener en cuenta el coste de operación: Los costes de refrigeración en las unidades flash y los costes de electricidad en general. Como se puede observar en la tabla 8, los costes de operación son muy importantes ya que suponen la mayor parte del coste anual. Es importante destacar que los costes de operación son costes variables que dependen en gran medida del volumen de producción. Por lo tanto, van alineados con los ingresos obtenidos.

### 6.1.5. Simulación de la producción de anhídrido acético

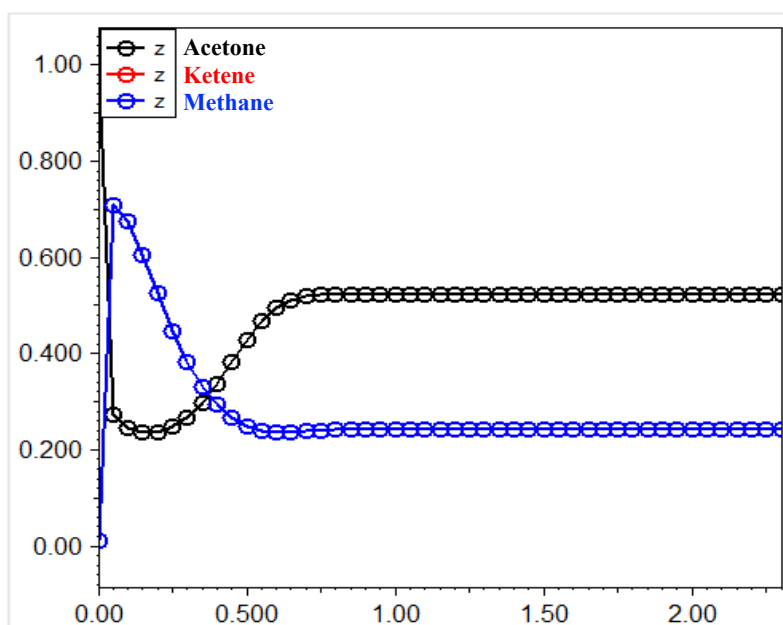
Una vez simulada la producción de anhídrido acético se obtienen los siguientes resultados.

**Tabla 9.** Resultados simulación PFR en EMSO.

Composición Acetona (fracción molar)	Composición Keteno (fracción molar)	Composición Metano (fracción molar)
0,519	0,240	0,240

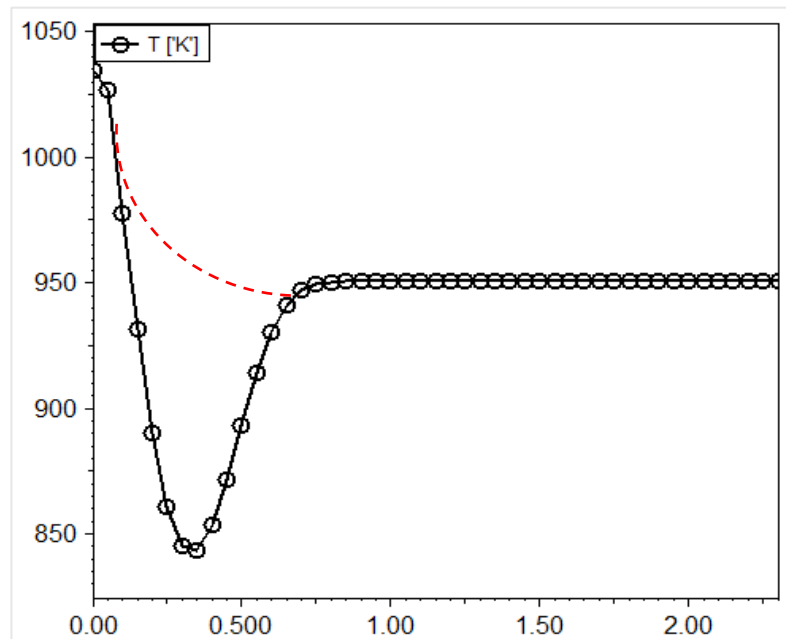
Conversión	Temperatura (K)
0,309	951

Como se puede observar se obtiene una conversión bastante baja, por lo que disminuye discretamente la composición de acetona y, a su vez, aumentan las composiciones de keteno y metano. En las siguientes gráficas se estudia como varía las composiciones y la temperatura a lo largo del reactor.



**Figura 38.** Composiciones de los componentes frente a la longitud del reactor en metros.

La figura 38 representa las composiciones de los componentes que intervienen en la reacción en fracción molar frente a la longitud del reactor en metros. Como se puede observar hay una pequeña desviación en los primeros valores del eje x, esto puede ser debido al método de discretización que utiliza EMSO para resolver las ecuaciones diferenciales. En líneas generales, se observa que las composiciones de keteno y de metano van aumentando gradualmente a la vez que disminuye la composición de acetona a lo largo del reactor hasta alcanzar unos valores estables, a partir de los 0.75 metros.



**Figura 39.** Temperatura del reactor en Kelvin frente a la longitud del reactor en metros.

La figura 39 representa la variación de la temperatura a lo largo del reactor. Al igual que en la figura anterior, se deben descartar los primeros valores del eje x ya que la disminución exponencial de la temperatura es debido al método de resolución que utiliza EMSO. Por lo general, se puede observar que la temperatura va disminuyendo hasta alcanzar valores estables. La temperatura disminuye debido a que se trata de una reacción endotérmica que tiene lugar en un reactor adiabático.

## 6.2. Comparación resultados en Matlab y EMSO

MATLAB (Matrix Laboratory) es un lenguaje de programación patentado de múltiples paradigmas y un entorno informático numérico desarrollado por MathWorks. MATLAB permite manipulaciones de matrices, trazado de funciones y datos, implementación de algoritmos, creación de interfaces de usuario e interactuar con programas escritos en otros lenguajes.

Con el objetivo de comparar el programa Matlab con EMSO se ha llevado a cabo la simulación de tres reactores continuos de tanque agitado en serie, este caso de estudio se ha realizado en el apartado anterior (5.6.2.) con EMSO.

Teniendo en cuenta los datos del problema realizado en EMSO, se lleva a cabo la siguiente simulación. Mediante el uso de dos scripts, el script función y el script principal donde se ejecuta la optimización.

```
function CAoutletReactor3 = simulate_CSTRs_in_seriesI(volume)
%
% reaction A --> products
reactionOrder = 1.5; % [dimensionless]
k = 1; % [m^{1.5}·kmol·m^{0.5}·h^{-1}]

% Feed stream
Feed_Volumetricflowrate = 2; % [m^3·h^{-1}]
Feed_concentration = 1; % [kmol·m^{-3}]

% Reactor data
nReactors = length(volume);

CAGuess = [0.5 0.5 0.5];
options = optimoptions('fsolve','Display','none');
CASolution = fsolve(@compute_molar_balance, CAGuess,options);
CAoutletReactor3 = CASolution(3);

format long % Result with more decimal places
CASolution

function molarBalance = compute_molar_balance(CA)
    for i = 1:nReactors
        if i == 1
            CAinlet(i) = Feed_concentration;
        else
            CAinlet(i) = CA(i-1);
        end

        reactionRateA(i) = -k * CA(i)^(reactionOrder);

        molarBalance(i) = Feed_Volumetricflowrate * (CAinlet(i) - CA(i)) ..
            + reactionRateA(i) * volume(i);
    end
end
end
```

Figura 40. Función objetivo para la simulación en Matlab.

En la función objetivo se describen las ecuaciones de los tres reactores continuos de tanque agitado, la velocidad de reacción, los balances molares y los datos del problema. En los datos del reactor ( $nReactors = \text{length}(\text{volumen})$ ) se especifica que habrá tantos reactores como componentes tenga el vector que el volumen de cada reactor, en este caso 3.



```

clear all
clc
close all

% Reactor data
volume = [2 2 2]; %[m^3] initial volume

%% Simulation given the volume of the reactors
CAoutletReactor3 = simulate_CSTRs_in_seriesI(volume)

%% Optimization

totalVolume = 7; %[m^3]
volumeGuess = [1 1 1]; %[m^3]
volumeLowerBound = [0 0 0]; %[m^3]
volumeUpperBound = [20 20 20]; %[m^3]

[optimalVolume, optimalCAoutletReactor3] = fmincon(...
    @simulate_CSTRs_in_series, volumeGuess, ...
    [1 1 1], totalVolume, [], [], ...
    volumeLowerBound, volumeUpperBound)

```

**Figura 41.** Script principal de la simulación de 3 CSTRs en serie con Matlab.

En el script principal se especifican el vector del volumen inicial, el rango de valores en el que se puede mover el vector volumen y finalmente se optimiza la función objetivo, sujeto a la restricción lineal de desigualdad que delimita el volumen total a 7 m<sup>3</sup> (esta se especifica con los argumentos situados en la posición 3 y 4 del comando «fmincon»).

Una vez realizadas las simulaciones en Matlab y en EMSO, se obtienen los resultados expuestos en la tabla 10 donde se puede observar una cierta desviación entre ambos resultados, teniendo en cuenta que las dimensiones de los reactores son iguales en ambas simulaciones.

**Tabla 10.** Comparativa de los resultados de la simulación 3 CSTRs en serie con Matlab y EMSO.

Simulación		EMSO	Matlab	Error (%)
Reactor	V (m <sup>3</sup> )	Ca (kmol/m <sup>3</sup> )	Ca (kmol/m <sup>3</sup> )	
CSTR 1	2,00	0,5670	0,570	0,0000%
CSTR 2	2,00	0,357	0,357	0,0000%
CSTR 3	2,00	0,240	0,240	0,0000%

El error entre los valores obtenidos en EMSO y en Matlab se calcula con la siguiente ecuación:

$$Error (\%) = \frac{|Valor_1 - Valor_2|}{Valor_1} 100 \quad (36)$$

Se puede observar en la tabla 10 que al simular en Matlab y en EMSO, fijando los mismos valores en los volúmenes de los reactores, se obtienen los mismos resultados sin apenas variaciones considerando que se han valorado los resultados a 6 cifras decimales. Por lo tanto, se puede afirmar que los resultados son exactos.

**Tabla 11.** Comparativa de los resultados de la optimización 3 CSTRs en serie con Matlab y EMSO.

Optimización	EMSO	Matlab	
Reactor	Ca (kmol/m <sup>3</sup> )	Ca (kmol/m <sup>3</sup> )	Error (%)
CSTR 1	0,562	0,562	0,0000%
CSTR 2	0,336	0,336	0,0003%
CSTR 3	0,210	0,210	0,0000%

Optimización	EMSO	Matlab	
Reactor	V (m <sup>3</sup> )	V (m <sup>3</sup> )	Error (%)
CSTR 1	2,078	2,078	0,0005%
CSTR 2	2,332	2,332	0,0004%
CSTR 3	2,591	2,591	0,0004%

Por otro lado, en las tablas 11 se pueden observar los resultados obtenidos en la optimización en Matlab y en EMSO de los tres reactores continuos de tanque agitado en serie. En la tabla superior están las concentraciones de cada reactor y en la tabla inferior los volúmenes de los reactores optimizados. Como se puede observar hay una ligera diferencia entre los resultados de la concentración del reactor 2 y en los resultados de los volúmenes optimizados, pero en líneas generales los errores son apenas apreciables, por lo que se puede afirmar que ambos programas proporcionan el mismo resultado.

Profile Summary				
Generated 17-Jun-2021 12:07:14 using performance time.				
Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
<a href="#">CSTRs_in_series_MAINI</a>	1	5.893 s	1.188 s	
<a href="#">fmincon</a>	1	3.148 s	1.145 s	
<a href="#">simulate_CSTRs_in_series</a>	72	2.383 s	0.798 s	
<a href="#">finDiffEvalAndChkErr</a>	1098	2.137 s	0.099 s	
<a href="#">barrier</a>	1	1.835 s	0.202 s	
<a href="#">fsolve</a>	73	1.572 s	0.252 s	
<a href="#">simulate_CSTRs_in_seriesI</a>	1	1.421 s	0.038 s	
<a href="#">optimoptions</a>	73	1.397 s	0.017 s	
<a href="#">createSolverOptions</a>	73	1.379 s	0.185 s	
<a href="#">optim\private\trustnleqn</a>	73	0.938 s	0.300 s	
<a href="#">findSubClasses</a>	1	0.692 s	0.616 s	
<a href="#">...Rs_in_series&gt;compute_molar_balance</a>	1376	0.669 s	0.669 s	
<a href="#">optimget</a>	1792	0.376 s	0.212 s	
<a href="#">createExitMsg</a>	1	0.220 s	0.198 s	
<a href="#">Fsolve&gt;Fsolve.Fsolve</a>	73	0.219 s	0.008 s	

Figura 42. Tiempo de resolución de 3 CSTRs en serie en Matlab.

En cuanto al tiempo de optimización, como se puede observar en la figura 42, Matlab tarda un total de 5,893 segundos en optimizar y simular los reactores en serie. Mientras que EMSO tarda 0,015 segundos en simular y 0,031 segundos en optimizar, un total de 0,046 segundos. Aunque el tiempo total que tardan ambos programas en resolver el problema es mínimo, es evidente que existe una gran diferencia entre ambos valores siendo el tiempo en EMSO muy inferior al tiempo que tarda Matlab en simular y optimizar el proceso.

En conclusión, EMSO tiene la gran ventaja de poseer una base de datos con los equipos más utilizados en la industria química, luego no habría que escribir los modelos de los equipos. Como resultado, es mucho más rápido simular un proceso químico en EMSO que en Matlab ya que se ahorra bastante tiempo, además los resultados obtenidos en ambos procesos no difieren apenas entre sí.

## 6.4. Comparación resultados en GAMS y EMSO

El programa GAMS (General Algebraic Modeling System) es un software de modelado de alto nivel para la optimización matemática. Está diseñado para modelar y resolver problemas de optimización de enteros mixtos, lineales y no lineales. El sistema está diseñado para aplicaciones de modelado complejas a gran escala y permite al usuario construir grandes modelos que se pueden adaptar a nuevas situaciones.

Con el objetivo de comparar la capacidad de resolución de EMSO con respecto a GAMS, se ha realizado el caso de estudio de planificación de la producción para comparar los resultados obtenidos.

En la siguiente figura se puede observar el problema de planificación de la producción planteado en GAMS, maximizando los beneficios de la producción. Cabe mencionar que EMSO utiliza un solver con diferenciación automática, al igual que GAMS.

```
VARIABLES
P valor de la funcion objetivo;

POSITIVE VARIABLES
xA1, xB0, xB1, xB2, xB3, xC2, xC3, xCD1, xCD2 toneladas_hora;

BINARY VARIABLES
yA1, yB0, yB2, yB3, yD1, yD2 binarias de cada proceso;

EQUATIONS
funobj, eq1, eq2, eq3, eq4, eq5, eq6, eq7, eq8, eq9, eq10, eq11,
eq12, eq13, disy, disy1, disy2;

funobj.. P =E= 100*(18*xCd1 + 180*yD2 + 15*(xCd2 - 10*yD2)
- 10*yA1 - 7,5*xA1 - 9,5*xB0 - 15*yB2 - 4*xB2 - 20*yB3 - 5,5*xB3);

eq1.. xB1 - 0,9*xA1 =E= 0
eq2.. xC2 - 0,82*xB2 =E= 0
eq3.. xC2 - xC3 - 15 =L= 0
eq4.. xC3 - 0,95*xB3 =L= 0
eq5.. xA1 - 16*yA1 =L= 0
eq6.. xB2 - 30*yB2 =L= 0
eq7.. xCD1 - 10*yD1 =L= 0
eq8.. xB2 + xB3 - xB1 - xB0 =E= 0
eq9.. xCD1 + xCD2 - xC2 - xC3 =E= 0
eq10.. xB0 - 50*yB0 =L= 0
eq11.. xB3 - 30*yB3 =L= 0
eq12.. xCD2 - 15*yD2 =L= 0
eq13.. 10*yD2 - xCD2 =L= 0

disy.. yD1 + yD2 - 1 =E= 0
disy1.. yA1 + yB0 - 2 =L= 0
disy2.. yB2 + yB3 - 1 =E= 0

MODELS
Tomadecisiones /all/;

SOLVE
tomadecisiones using MIP maximizing P;
```

Figura 43. Planificación de la producción en GAMS.

En la siguiente figura se observa los resultados obtenidos en GAMS, a la izquierda de la tabla se encuentran las variables y en la columna level los valores finales de cada variable.

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR P	-INF	600.0000	+INF	.
---- VAR xA1	.	16.0000	+INF	.
---- VAR xB0	.	.	+INF	-75.0000
---- VAR xB1	.	14.4000	+INF	.
---- VAR xB2	.	.	+INF	-45.0000
---- VAR xB3	.	14.4000	+INF	.
---- VAR xC2	.	.	+INF	.
---- VAR xC3	.	13.6800	+INF	.
---- VAR xCD1	.	.	+INF	.
---- VAR xCD2	.	13.6800	+INF	.
---- VAR yA1	.	1.0000	1.0000	-400.0000
---- VAR yB0	.	1.0000	1.0000	EPS
---- VAR yB2	.	.	1.0000	-1500.0000
---- VAR yB3	.	1.0000	1.0000	-2000.0000
---- VAR yD1	.	.	1.0000	3000.0000
---- VAR yD2	.	1.0000	1.0000	3000.0000

P valor de la funcion objetivo  
 xCD2 toneladas\_hora  
 yD2 binarias de cada proceso

Figura 44. Resultados en GAMS de la planificación de la producción.

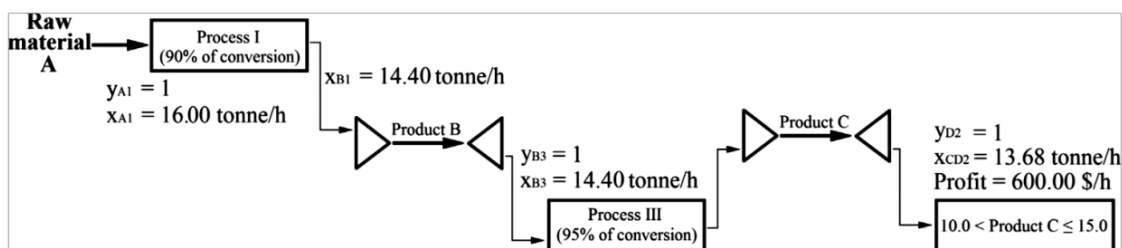


Figura 45. Resultados de la planificación de la producción en EMSO.

Como se puede observar en ambos programas, EMSO y GAMS, se obtienen los mismos resultados optando por el mismo camino ambos resultados, por los mismos procesos de producción, y los mismos valores de las variables.

```

**** REPORT SUMMARY :      0  NONOPT
                           0  INFEASIBLE
                           0  UNBOUNDED
                           5  PROJECTED

EXECUTION TIME      =      0.420 SECONDS      3 MB  35.1.0  r82a9585  DEX-DEG
  
```

Figura 46. Tiempo de optimización de la producción en GAMS.

En cuanto al tiempo, el programa EMSO tarda 0,0 segundos en simular y 0,735 segundos en optimizar la planificación de la producción. Por otro lado, el programa GAMS tarda un total de 0,420 segundos en simular y optimizar la planificación de la producción. Como se puede observar ambas cifras son mínimas, ya que no llegan al segundo. Por consiguiente, se puede afirmar que EMSO es un programa que no solamente aborda problemas complejos de simulación y optimización orientados a la industria química, sino que, además, es capaz de competir con un programa tan competente y presente en el mercado como lo es GAMS.

## 6.5. Comparación resultados en Matlab, Aspen Hysys y EMSO

Con el objetivo de comparar la capacidad de resolución de ecuaciones diferenciales en EMSO con respecto a otros simuladores de procesos presentes en el mercado. En este apartado se llevará a cabo la simulación de la producción de anhídrido acético en el programa Matlab y en Aspen Hysys. Con la finalidad de compara los resultados.

Se lleva a cabo la siguiente simulación en Matlab, con los datos del problema de la producción de anhídrido acético realizado en EMSO. Mediante el uso de dos scripts, el script función donde se localizan las ecuaciones diferenciales del balance molar y del balance de energía, y el script principal donde se ejecuta el comando ODE45 de Matlab para resolver ecuaciones diferenciales.

```
function dydL=fPFR_Acetone(L,y)
X = y(1); %Conversion of Acetone
T = y(2); %Temperature (K)

global alpha F Tin Tref Composition_in Area Mm P R Cp AHref
% Reaction Acetone --> Ketene + Methane
%Mole streams
Mm_in = Composition_in * Mm';

ntot_in = F/Mm_in; %kmol/h
nj_in = ntot_in .* Composition_in; %kmol/h
nAcetone = nj_in(1) * (1 - X); %kmol/h
nj = [nAcetone (nj_in(1)*X+nj_in(2)) (nj_in(1)*X+nj_in(3))]; %kmol/h
ntot = nj(:,1)+ nj(:,2)+ nj(:,3); %kmol/h

%Compositions
Composition_out = [nj(:,1)./ntot...
    nj(:,2)./ntot nj(:,3)./ntot]; %Mole fraction

%Volumetric flow rate
% PQv = nRT
Qv_in = ntot_in * R * Tin/(P * 101.325) %m3/h

Qv = Qv_in .* (T/Tin) .* (1 +...
    (Composition_in(1)*sum(alpha)/(-alpha(1))) *X) %m3/h

%Rate reaction
Cj = (1/Qv).*nj %kmol/m3

k = exp(34.34-(34222/T))*3600; %1/h
rj = k .* Cj; %kmol/m3 h

%%Mole balance
dXdL = -(Area/nj_in(1))*sum(rj * alpha');

%%Energy balance
AHi = AHref + sum(Cp*alpha')*(T-Tref);
dTdL = -sum(AHi*rj)*Area/sum(nj*Cp');

dydL = [dXdL; dTdL];

end
```

Figura 47. Script fPFR\_Acetone en Matlab.

En la función objetivo se describen las ecuaciones diferenciales del balance molar y el balance de energía. Además, al tratarse de gases se calcula el flujo volumétrico para cada tramo del reactor. Se especifica la cinética de la reacción, la variable independiente (la longitud del reactor) y las variables dependientes, la conversión y la temperatura.

```
close all
clc
clear all

global alpha F Tin Tref Composition_in Area Mm P R Cp AHref

%DATA
% Acetone --> Ketene + Methane
alpha = [-1 1 1];
Mm = [58 42 16]; %kg/kmol
Cp = [162.09 84.71 73.75]; %kJ/kmol K
F = 8000; %kg/h
Tin = 1035;%K
Tref = 298; %K
Composition_in = [0.98 0.01 0.01]; %mole fraction
Area = 1.27/2.28; %m2
P = 1.6; %atm (constant P)
R = 8.32; %kJ/kmol K
AHref = 80770; %kJ/kmol

%CALL ODE45
Lrange = [0 2.28]; %m
y_initial = [0 Tin];

[L,y] = ode45(@fPFR_Acetone, Lrange, y_initial);
```

*Figura 48. Script PFR\_Acetone MAIN en Matlab.*

A continuación, mediante el script MAIN se especifican todos los datos del problema, los valores iniciales de las variables dependientes y el límite de la variable independiente, en este caso la longitud final del reactor 2.28 m. Una vez se han especificado todos los datos se ejecuta el comando ODE45 de Matlab para resolver ecuaciones diferenciales.

Es importante destacar que los valores de los calores específicos  $C_{pj}$  se han obtenido a través del Instituto Nacional de Estándares y Tecnología (NIST) del departamento de comercio de EE.UU. (UU., 2018).

En este apartado sólo se presentarán los resultados de la simulación llevada a cabo en Aspen Hysys, pero en el anexo II se puede encontrar todo el procedimiento detallado para llevar a cabo la simulación.

Una vez realizadas las simulaciones en Matlab, Aspen Hysys y en EMSO, se obtienen los resultados expuestos en las siguientes tablas donde se pueden observar diferencias evidentes entre los resultados.

**Tabla 12.** Comparación conversiones finales Matlab-Hysys.

	<b>Matlab</b>	<b>Hysys</b>	<b>Error (%)</b>
<b>Conversión</b>	0,149	0,192	22,3%

Como se puede observar en la tabla 12 hay una cierta diferencia entre la conversión final obtenidas en Matlab con respecto a la obtenida en Aspen Hysys. Esto puede ser debido a que en Matlab se ha considerado que se trabaja con gas ideal. Además, no se tienen en cuenta todas las propiedades físicas de los compuestos que intervienen en la reacción mientras que Hysys tiene una base de datos con todas las propiedades físicas de los compuestos.

**Tabla 13.** Comparación composiciones finales de los simuladores con el simulador EMSO.

<b>Componente</b>	<b>EMSO</b>	<b>Matlab</b>	<b>Error (%)</b>	<b>EMSO</b>	<b>Hysys</b>	<b>Error (%)</b>
Acetona	0,520	0,728	40,0%	0,520	0,667	28,3%
Keteno	0,240	0,136	43,3%	0,240	0,166	30,7%
Metano	0,240	0,136	43,3%	0,240	0,166	30,7%

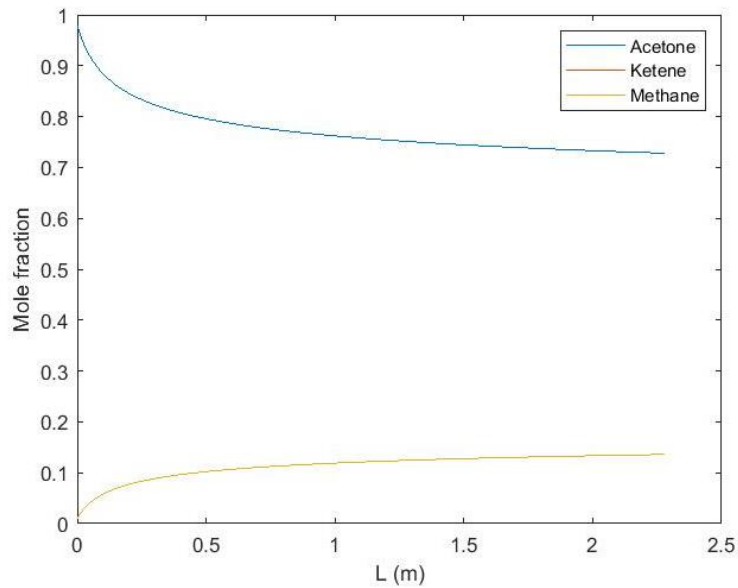
Por otro lado, al comparar las composiciones obtenidas en Matlab y Hysys con respecto a EMSO aumenta ampliamente la diferencia entre los resultados obtenidos. Como se ha explicado en el apartado de simulación de la producción de anhídrido acético, EMSO utiliza un método de discretización para resolver las ecuaciones diferenciales. Es un método muy sencillo, pero que supone una menor precisión en los resultados. Del mismo modo, es importante resaltar que EMSO, al igual que Hysys, dispone de una base de datos con todas las propiedades físicas de los compuestos que intervienen en la reacción.

**Tabla 14.** Comparación de la conversión y la temperatura final entre los simuladores.

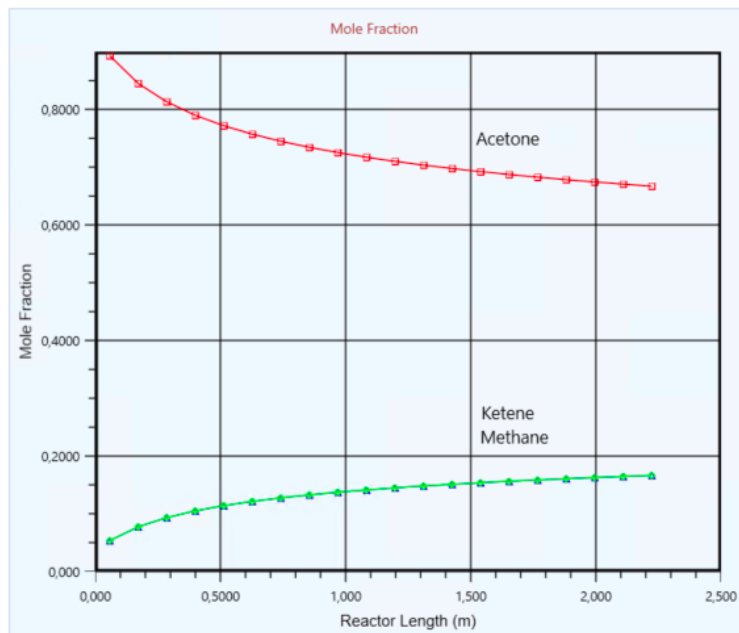
	<b>Conversión</b>	<b>EMSO</b>	<b>Error %</b>	<b>T (K)</b>	<b>EMSO</b>	<b>Error %</b>
<b>Hysys</b>	0,192	0,309	38,0%	937	951	1,46%
<b>Matlab</b>	0,149	0,309	51,8%	928	951	2,43%

Al comparar las temperaturas a la salida del reactor se puede observar que hay muy poca diferencia entre si. En cuanto a las conversiones finales, evidentemente se observa la misma tendencia que las composiciones finales, siendo muy cercanos los valores entre Matlab y Hysys, y alejándose algo más el valor obtenido mediante EMSO.



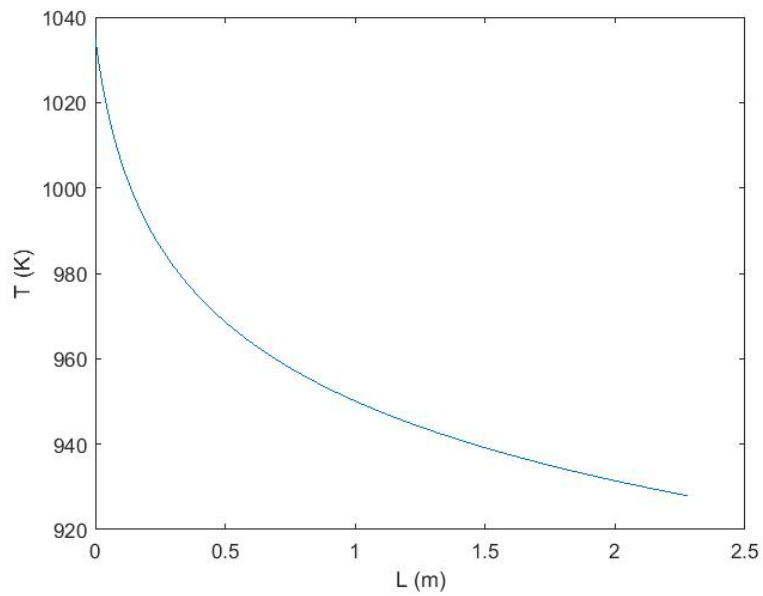


**Figura 49.** Composiciones frente a la longitud del reactor en Matlab.

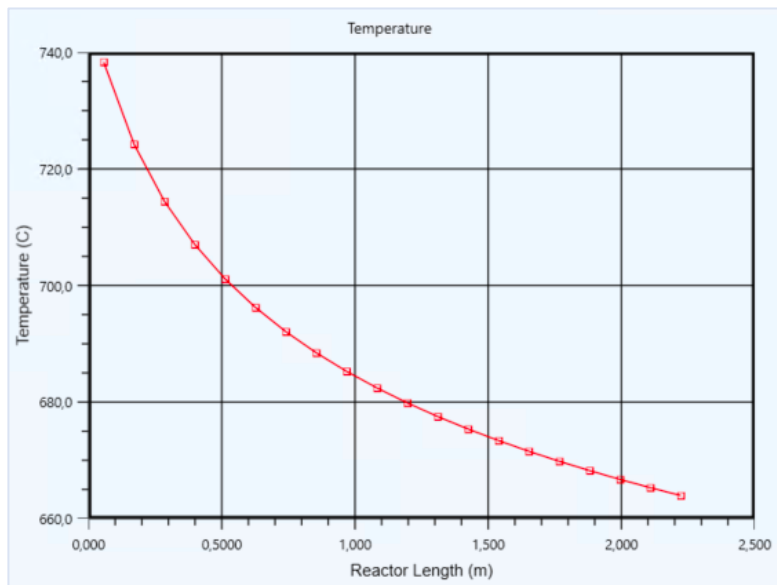


**Figura 50.** Composiciones frente a la longitud del reactor en Hysys.

Las gráficas de las composiciones frente a la longitud del reactor obtenidas en Matlab y Hysys siguen la misma tendencia que la figura 38. Se observa que las composiciones de keteno y de metano van aumentando gradualmente, a la vez que disminuye la composición de acetona a lo largo del reactor hasta alcanzar unos valores estables, a partir de los 0.75 metros aproximadamente.



**Figura 51.** Temperatura frente a la longitud del reactor en Matlab.



**Figura 52.** Temperatura frente a la longitud del reactor en Hysys.

Las gráficas de la temperatura frente a la longitud del reactor obtenidas en Matlab y Hysys siguen la misma tendencia que la figura 39, gráfica obtenida en EMSO. Donde se puede observar que la temperatura va disminuyendo hasta alcanzar valores estables.

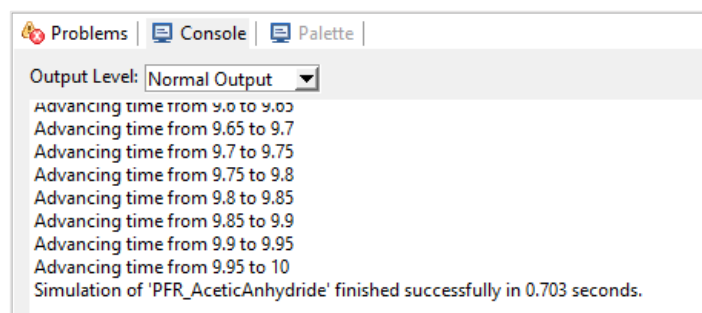


Figura 53. Tiempo de resolución del PFR en EMSO.

Profile Summary				
Generated 08-Jul-2021 19:43:12 using performance time.				
Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
<a href="#">PFR_AcetoneMAIN</a>	1	6.153 s	0.519 s	
<a href="#">legend</a>	1	3.296 s	0.122 s	
<a href="#">legend&gt;make_legend</a>	1	3.172 s	0.087 s	
<a href="#">Legend.Legend&gt;Legend.Legend</a>	1	1.747 s	0.685 s	
<a href="#">ode45</a>	1	1.321 s	0.087 s	
<a href="#">funfun\private\odearguments</a>	1	0.961 s	0.914 s	
<a href="#">Legend.Legend&gt;Legend.setAxes</a>	1	0.788 s	0.001 s	
<a href="#">Legend.Legend&gt;Legend.setAxes_1</a>	1	0.786 s	0.001 s	
<a href="#">Legend.Legend&gt;Legend.setAxesImpl</a>	1	0.784 s	0.013 s	
<a href="#">legendcolorbarlayout</a>	1	0.761 s	0.145 s	

Figura 54. Tiempo de resolución de la simulación PFR en Matlab.

En cuanto al tiempo de optimización, como se puede observar en las figuras anteriores, Matlab tarda un total de 6,153 segundos en resolver las ecuaciones diferenciales y simular el reactor. Mientras que EMSO tarda un total de 0,703 segundos en simular el proceso. Ambos simuladores tardan segundos en resolver las ecuaciones diferenciales, pero es evidente que Matlab tarda casi 9 veces más en resolver la simulación que EMSO. Debido principalmente a la complejidad del método de resolución que utiliza cada programa.

Es importante destacar que el comando ODE45 de Matlab utiliza el método de Dormand-Prince, un método para la resolución de ecuaciones diferenciales ordinarias. Pertenece a la familia de métodos Runge-Kutta (ODE45).

## 7. Conclusiones

Se ha llevado a cabo una investigación sobre el manejo del programa EMSO, donde se ha podido resolver una amplia variedad de problemas. El funcionamiento del programa en sí ha supuesto un reto, debido principalmente a la poca información que hay al respecto. Al ser un programa brasileño creado principalmente para uso propio de sus creadores, no con el objetivo de comercializar dicho programa, no hay muchas fuentes de información donde poder consultar dudas o ejemplos, aparte de la página web de EMSO. Por otro lado, la base de datos de EMSO cuenta con una gran cantidad de ejemplos muy educativos que facilitan mucho el aprendizaje del programa, teniendo la posibilidad de partir de un ejemplo hecho en el programa y modificarlo según las condiciones del problema que se desea resolver.

Después de realizar una amplia variedad de simulaciones en EMSO, se puede destacar la rapidez en la que el programa resuelve simulaciones que en algunos casos son complejas. Además, la verificación y conversión automática de las unidades de medida, es sin duda una de sus grandes ventajas, ya que no hay que preocuparse por si se ha cometido un error en las unidades. Al ser un programa basado en ecuaciones uno de los temas que más preocupa es si se tendrán que reescribir todas las ecuaciones que representan los equipos más comunes de la industria química cada vez que se simula un proceso, pero gracias a la base de datos del programa esto no es un problema ya que están definidos la mayoría de los equipos más utilizados en la industria. Asimismo, EMSO utiliza un lenguaje orientado a objetos, lo que permite al usuario desarrollar modelos complejos componiéndolos con modelos pequeños existentes o desarrollar modelos específicos a partir de modelos estándar. Otra gran ventaja del programa es la diferenciación simbólica y automática que es capaz de diferenciar los distintos comandos, números y formatos mediante colores que se incorporan automáticamente al escribir. El simulador de procesos EMSO es una herramienta muy útil y flexible, tanto en el ámbito académico como en el ámbito profesional.

Comparado con otros programas como Aspen Hysys, al ser EMSO un programa basado en ecuaciones es muy versátil a la hora de editar un equipo y optimizar procesos, es mucho más rápido y sencillo que optimizar procesos complejos en un simulador de procesos secuencial-modular como Aspen Hysys. Por otro lado, comparado con Matlab, EMSO tiene la ventaja de poseer una base de datos de los equipos más utilizados en la industria química, al ser un programa diseñado especialmente por ingenieros químicos no habría que escribir los modelos de los equipos. Por lo tanto, es mucho más rápido simular un proceso químico en EMSO que en Matlab ya que se ahorra bastante tiempo, además los resultados obtenidos en ambos procesos no difieren apenas entre sí. Al realizar la comparación de EMSO con GAMS se puede destacar la similitud de los resultados, en el caso de estudio de planificación de la producción se obtienen exactamente los mismos resultados. En cuanto al tiempo de resolución, se observa un ligero incremento con respecto al tiempo en GAMS, pero ambos valores siguen siendo mínimos.

Finalmente, teniendo en cuenta todas las ventajas que presenta el simulador de procesos EMSO orientado a ecuaciones se puede decir que es un programa fiable, competente, rápido y muy versátil para la simulación y optimización de procesos químicos, además es gratuito para las instituciones académicas. Igualmente, puede ser muy útil también en otros campos como la economía, la administración, la ingeniería y la ciencia.

## 8. Apéndices y anexos

Apéndice I: Tablas resumen de las referencias más utilizadas en EMSO.

## Introduction

EMSO is the acronym for Environment for Modeling, Simulation, and Optimization. It is a dynamic simulator for general processes.

This is the EMSO quick reference card, for detailed documentation please consult the EMSO User's Guide.

You can find the latest version of this document, the EMSO User's Guide, and the software itself at <http://www.enq.ufrgs.br/alsoc>.

## EMSO Types – types.mso

Name	Base	Default	Lower	Upper	Unit
Real	Built-In	1	$-\infty$	$\infty$	<i>none</i>
coefficient	Real	1	-50	50	<i>none</i>
Constant	Real	10	-5e3	5e3	<i>none</i>
positive	Real	1.0	-1e-6	$\infty$	<i>none</i>
negative	Real	-1.0	$-\infty$	1e-6	<i>none</i>
fraction	Real	0.5	-1e-6	1.00001	<i>none</i>
percent	Real	50	0	100	<i>none</i>
control_signal	Real	1	-1e9	1e9	<i>none</i>
efficiency	Real	0.5	0	1	<i>none</i>
pressure	Real	1	1e-30	5e7	<i>atm</i>
press_delta	pressure	0.01	-5e6	5e7	<i>atm</i>
head_mass	Real	50	-1e6	1e6	<i>kJ/kg</i>
head	Real	50	-1e6	1e6	<i>kJ/kmol</i>
temperature	Real	300	27	5273	<i>K</i>
temp_delta	temperature	30	-1e3	1e3	<i>K</i>
time_h	positive	1	-1e-6	1e4	<i>h</i>
time_min	time_h	1	-1e-6	$\infty$	<i>min</i>
time_sec	time_h	1	-1e-6	$\infty$	<i>s</i>
frequency	positive	1	-1e-6	100	<i>1/s</i>
angle	Real	0	-7	7	<i>rad</i>
area	positive	1	-1e-6	1e6	<i>m<sup>2</sup></i>
length	positive	1	-1e-6	5e6	<i>m</i>
length_delta	length	1	-1e3	$\infty$	<i>m</i>
volume	positive	10	-1e-6	1e3	<i>m<sup>3</sup></i>
volume_mol	positive	10	-1e-6	1e6	<i>m<sup>3</sup>/mol</i>

Name	Base	Default	Lower	Upper	Unit
volume_mass	positive	10	-1e-6	1e30	<i>m<sup>3</sup>/kg</i>
current	positive	10	-1e-6	1e12	<i>A</i>
charge	Real	10	-1e12	1e12	<i>C</i>
capacitance	positive	10	-1e-6	1e12	<i>F</i>
indutance	positive	10	-1e-6	1e12	<i>V * s/A</i>
voltage	positive	100	-1e-6	1e9	<i>V</i>
resistance	positive	50	-1e-6	800	<i>ohm</i>
potency	Real	10	-1e3	1e3	<i>kW</i>
currency	Real	0	$-\infty$	$\infty$	<i>US\$</i>
mass	positive	2.5	-1e-6	1e6	<i>kg</i>
mol	positive	2.5	-1e-6	1e6	<i>kmol</i>
molweight	Real	75	1	1e8	<i>kg/kmol</i>
molweight_inv	Real	1	0	1e30	<i>kmol/kg</i>
dens_mol	Real	1	1e-30	5e3	<i>kmol/m<sup>3</sup></i>
dens_mass	Real	1e3	1e-30	5e5	<i>kg/m<sup>3</sup></i>
conc_mol	dens_mol	1e-3	1e-30	5e3	<i>kmol/m<sup>3</sup></i>
inv_conc_mol	Real	0.05	2e-4	1e30	<i>m<sup>3</sup>/kmol</i>
conc_mass	dens_mass	1e3	1e-30	5e3	<i>kg/m<sup>3</sup></i>
inv_conc_mass	Real	1e-3	2e-6	1e30	<i>m<sup>3</sup>/kg</i>
reaction_mol	Real	10	-1e6	1e6	<i>kmol/h/m<sup>3</sup></i>
reaction_mass	Real	1e3	-1e3	1e9	<i>kg/h/m<sup>3</sup></i>
cp_mass	Real	1.0	0.1	10.0	<i>kJ/kg/K</i>
cp_mol	Real	100	1	1e3	<i>J/mol/K</i>
cv_mol	Real	100	1	1e3	<i>J/mol/K</i>
enth_mass	Real	500	-1e4	1e4	<i>kJ/kg</i>
enth_mol	Real	500	-1e9	1e9	<i>J/mol</i>
entr_mol	Real	0	-1e4	1e4	<i>kJ/kmol/K</i>
entr_mass	Real	0	-1e4	1e4	<i>kJ/kg/K</i>
heat_reaction	Real	1e3	-1e8	1e8	<i>kJ/kmol</i>
heat_rate	Real	1e3	-1e11	1e11	<i>J/s</i>
heat_flux	Real	1	-1e5	1e5	<i>kW/m<sup>2</sup></i>
heat_trans.coeff	positive	1	-1e-6	1e3	<i>kW/m<sup>2</sup>/K</i>
energy	Real	1e4	-1e11	1e11	<i>kJ</i>
energy_mass	Real	1e4	-1e15	1e15	<i>kJ/kg</i>
energy_mol	Real	1e4	-1e15	1e15	<i>kJ/kmol</i>
power	Real	10	-1e8	1e8	<i>kW</i>
flow_mass	positive	1e3	-1e-6	1e10	<i>kg/h</i>

Name	Base	Default	Lower	Upper	Unit
flow_mass_delta	flow_mass	1e3	-1e10	1e10	kg/h
flow_vol	positive	1	-1e-6	1e12	m <sup>3</sup> /h
flow_vol_delta	flow_vol	1	-1e12	1e12	m <sup>3</sup> /h
flow_mol	positive	10	-1e-6	1e8	kmol/h
flow_mol_delta	flow_mol	10	-1e8	1e8	kmol/h
flux_mol	positive	1	-1e-6	1e4	kmol/s/m <sup>2</sup>
flux_mol_delta	flux_mol	1	-1e4	1e4	kmol/s/m <sup>2</sup>
flux_mass	positive	1	-1e-6	1e6	kg/s/m <sup>2</sup>
flux_mass_delta	flux_mass	1	-1e6	1e6	kg/s/m <sup>2</sup>
flux_vol	positive	1	-1e-6	1e4	m <sup>3</sup> /s/m <sup>2</sup>
flux_vol_delta	flux_vol	1	-1e4	1e4	m <sup>3</sup> /s/m <sup>2</sup>
vel_angular	Real	1e3	-1e5	1e5	rpm
rotation	Real	5e2	-1e4	1e4	rad/s
velocity	Real	1	-1e5	1e5	m/s
velocity_delta	velocity	0	-1e3	-1e3	m/s
acceleration	Real	9.81	-1e3	1e3	m/s <sup>2</sup>
fricfactor	Real	0.05	1e-5	2e3	none
moment_inertia	Real	100	1e-3	1e4	kg * m <sup>2</sup>
hookes_const	Real	1e5	1	1e10	N/m
conductivity	Real	1.0	1e-3	500	W/m/K
diffusivity	positive	1e-3	-1e-6	1	cm <sup>2</sup> /s
fugacity	positive	0.5	-1e-6	10	none
viscosity	Real	1	1e-30	1e5	cP
vol_mol	volume_mol	10	-1e-6	1e6	m <sup>3</sup> /mol
vol_mass	volume_mass	10	-1e-6	1e6	m <sup>3</sup> /kg
spec_surface_vol	positive	1e5	-1e-6	1e15	m <sup>2</sup> /m <sup>3</sup>
spec_surface_mass	positive	100	-1e-6	1e12	m <sup>2</sup> /kg
surf_tens	positive	0.05	-1e-6	1	N/m
act_coeff	positive	1	-1e-6	30	none
ph	Real	7	-5	20	none

## Built-In Functions

Function	Meaning
abs(Z)	Returns the magnitude or absolute value of Z
max(Z)	Returns the maximum value of Z
min(Z)	Returns the minimum value of Z
sign(Z)	Returns the signal of Z (-1 if Z < 0 e 1 if Z > 0
round(Z)	Returns the small integer value of Z

Function	Meaning
sinh(Z)	Returns the hyperbolic sine of Z
cosh(Z)	Returns the hyperbolic cosine of Z
tanh(Z)	Returns the hyperbolic tangent of Z
coth(Z)	Returns the hyperbolic cotangent of Z
sin(Z)	Returns the sine of Z
cos(Z)	Returns the cosine of Z
tan(Z)	Returns the tangent of Z
asin(Z)	Returns the angle whose sine is Z
acos(Z)	Returns the angle whose cosine is Z
diff(Z)	Returns the derivative of Z with respect to time
atan(Z)	Returns the angle whose tangent is Z
sum(Z)	Returns the sum of components of a vector or matrix Z
prod(Z)	Returns the product of a vector or matrix Z
sumt(Z)	Returns the transpose sum of components of a vector or matrix Z
prodt(Z)	Returns the transpose product of components of a vector or matrix Z
transp(Z)	Returns the transpose of a matrix Z
exp(Z)	Returns the exponential function, e raised to the power Z
log(Z)	Returns the base 10 logarithm of Z
ln(Z)	Returns the natural logarithm (base e) of Z
sqrt(Z)	Returns the square root of Z

## Units Of Measurement (UOM)

### Fundamental Units

m	length in meters
kg	mass in kilogram
s	time in seconds
K	temperature in Kelvin
A	electric current in Ampere
mol	the amount of substance in mole
cd	the luminous intensity in Candela
rad	angle measure in radian
US\$	money in dollar (USA)

**Derived Units**

<b>Acceleration of Gravity</b>		
ga	$9.80665\text{ m/s}^2$	standard acceleration of gravity
<b>Angle</b>		
arcs	$4.8481368111\text{e-6 rad}$	arcsecond
arcmin	$2.90888208666\text{e-4 rad}$	arcminute
grad	$1.57079632679\text{e-2 rad}$	grad
deg	$1.74532925199\text{e-2 rad}$	degree
<b>Area</b>		
acre	$4046.87260987\text{ m}^2$	acre
a	$100\text{ m}^2$	are
ha	$10000\text{ m}^2$	hectare
b	$1\text{e-28 m}^2$	barn
<b>Electric</b>		
Wb	$\text{kg m}^2/\text{A s}^2$	weber
T	$\text{kg}/\text{A s}^2$	tesla
S	$\text{A}^2\text{ s}^3/\text{kg m}^2$	siemens
mho	$\text{A}^2\text{ s}^3/\text{kg m}^2$	mho
Fdy	$96487\text{ A s}$	faraday
F	$\text{A}^2\text{ s}^4/\text{kg m}^2$	farad
ohm	$\text{kg m}^2/\text{A}^2\text{ s}^3$	ohm
C	$\text{A s}$	unit of relative current for batteries
V	$\text{kg m}^2/\text{A s}^3$	volt
<b>Energy</b>		
J	$\text{kg m}^2/\text{s}^2$	joule
kJ	$1\text{e3 J}$	kilojoule
MJ	$1\text{e6 J}$	megajoule
GJ	$1\text{e9 J}$	gigajoule
eV	$1.60217733\text{e-19 J}$	electronvolt
MeV	$1\text{e6 eV}$	megaelectronvolt
therm	$105506000\text{ J}$	therm
Btu	$1055.05585262\text{ J}$	British thermal unit
cal	$4.1868\text{ J}$	calorie
kcal	$1\text{e3 cal}$	kilo calorie
erg	$1\text{e-7 J}$	erg
<b>Force</b>		
N	$\text{kg m/s}^2$	newton

pdl	$0.138254954376\text{ N}$	poundal
lbf	$4.44822161526\text{ N}$	pounds of force
kip	$4448.22161526\text{ N}$	kip
gf	$0.00980665\text{ N}$	gram force
kgf	$1\text{e3 gf}$	kilogram force
dyn	$0.00001\text{ N}$	dyne
<b>Length</b>		
cm	$1\text{e-2 m}$	centimeter
mm	$0.1\text{ cm}$	millimeter
fermi	$1\text{e-15 m}$	fermi
$\text{\AA}$	$1\text{e-10 m}$	angstrom
$\mu$	$1\text{e-6 m}$	micro
mil	$2.54\text{e-5 m}$	mil
ftUS	$0.304800609601\text{ m}$	international foot
fath	$1.82880365761\text{ m}$	fathom
rd	$5.02921005842\text{ m}$	rod
chain	$20.1168402337\text{ m}$	chain
miUS	$1609.34721869\text{ m}$	US statute miles
nmi	$1852\text{ m}$	nautical mile
mi	$1609.344\text{ m}$	International Mile
km	$1000\text{ m}$	Kilometer
au	$1.495979\text{e11 m}$	Astronomical Unit
lyr	$9.46052840488\text{e15 m}$	light year
pc	$3.08567818585\text{e16 m}$	parsec
Mpc	$3.08567818585\text{e22 m}$	megaparsec
in	$0.0254\text{ m}$	inch
ft	$0.3048\text{ m}$	foot
yd	$0.9144\text{ m}$	yard
<b>Mass</b>		
u	$1.6605402\text{e-27 kg}$	atomic mass unit
grain	$0.00006479891\text{ kg}$	grain
ct	$0.0002\text{ kg}$	carat
ozt	$0.0311034768\text{ kg}$	troy ounce
t	$1000\text{ kg}$	tonne
tonUK	$1016.0469088\text{ kg}$	ton (UK)
ton	$907.18474\text{ kg}$	ton
lbt	$0.3732417216\text{ kg}$	troy pound



slug	14.5939029372*kg	slug
oz	0.028349523125*kg	ounce
lb	0.45359237*kg	pound
g	kg/1000	gram
kmol	1e3*mol	kilomole
lbmol	453.59237*mol	pound mole
<b>Money</b>		
R\$	US\$/3.05	Brazilian money (Real)
<b>Power</b>		
W	$kg * m^2 / s^3$	watt
kW	1e3*W	Kilowatt
MW	1e6*W	megawatt
hp	745.699871582*W	horsepower
<b>Pressure</b>		
Pa	kg/m/s <sup>2</sup> pascal	
kPa	1e3*Pa	Kilopascal
MPa	1e3*kPa	megapascal
inH2O	248.84*Pa	inch of water column
inHg	3386.38815789*Pa	inch of mercury
mmHg	133.322368421*Pa	millimeter of mercury
torr	133.322368421*Pa	torr
psi	6894.75729317*Pa	pound per square inch
bar	1e5*Pa	bar
atm	101325*Pa	atmosphere
<b>Radiation</b>		
R	0.000258*A*s/kg	R
Ci	3.7e10/s	curie
Bq	1/s	becquerel
Sv	$m^2 / s^2$	sievert
rem	$0.01 * m^2 / s^2$	rem
Gy	$m^2 / s^2$	gray
<b>Temperature</b>		
degR	K/1.8	degree Rankine
<b>Time</b>		
Hz	1/s	hertz
min	60*s	minute
rpm	1/min	revolution per minute

h	60*min	hour
d	24*h	day
yr	31556925.9744*s	year
<b>Velocity</b>		
c	299792458*m/s	velocity (or "speed") of light
knot	0.514444444444*m/s	knot
mph	0.44704*m/s	mile per hour
kph	0.277777777777*m/s	kilometer per hour
<b>Viscosity</b>		
St	0.0001*m <sup>2</sup> /s	stoke
P	0.1*kg/m/s	poise
cP	0.001*kg/m/s	centipoise
<b>Volume</b>		
st	m <sup>3</sup>	Stere
fbm	0.002359737216*m <sup>3</sup>	board foot
pk	0.0088097675*m <sup>3</sup>	peck
bu	0.03523907*m <sup>3</sup>	bushel
bbl	0.158987291928*m <sup>3</sup>	barrel
trp	4.92892159375e-6*m <sup>3</sup>	teaspoon
tbsp	1.47867647813e-5*m <sup>3</sup>	tablespoon
ozUK	2.8413075e-5*m <sup>3</sup>	fluid ounce (UK)
ozfl	2.95735295625e-5*m <sup>3</sup>	fluid ounce
cu	2.365882365e-4*m <sup>3</sup>	US Cup
l	1e-3*m <sup>3</sup>	liter
ml	1e-3*l	milliliter
pt	0.000473176473*m <sup>3</sup>	pint
qt	0.000946352946*m <sup>3</sup>	quart
gal	0.00378541178*m <sup>3</sup>	gallon
galC	0.00454609*m <sup>3</sup>	imperial gallon
galUK	0.004546092*m <sup>3</sup>	gallon (UK)

## Modeling Language

### Conventions:

- Every command enclosed by [ ] is optional.
- The code a | b | c means a or b or c.
- Every command between < > is an emphasis in that command.
- FlowSheet accepts all commands of a Model
- Optimization and Estimation accept all commands of a FlowSheet

### The FlowSheet Entity

---

```
FlowSheet FlowSheetName [as base]
DEVICES
DevName as base [(Brief="brief")];
CONNECTIONS
DevName.varName<in> to DevName.varName<out>;
SET
ParName = <expression involving only parameters> | value ;
SPECIFY
["equation name"] DevName.varName = expression | value;
INITIAL
["equation name"] DevName.varName = expression | value;
OPTIONS
    Dynamic      = false | true;
    TimeStart    = value;
    TimeStep     = value;
    TimeEnd      = value;
    TimeUnit     = 'unit';
    SparseAlgebra = false | true;
    GuessFile    = "File Name";
    InitialFile  = "File Name";
NLASolver(File="sundials" | "nlasolver",
    RelativeAccuracy=1e-3, AbsoluteAccuracy=1e-6, MaxIterations =100);
DAESolver(File="sundials" | "dassl" | "dasslc" | "mebdf",
    RelativeAccuracy=1e-3, AbsoluteAccuracy=1e-6, EventAccuracy =1e-2);
NLPSolver(File="complex" | "ipopt_emsol", RelativeAccuracy=1e-3);
end
```

---

### The Model Entity

---

```
Model modelName [as base]
ATTRIBUTES
Icon      = "file name";
Palette   = false | true;
Brief     = "A shortcut information about the model";
Info      = "A detailed information about the model";
PARAMETERS
[out] ParName;
[out] ParName as base;
[out] ParName as base (Brief="brief", Default=value, Lower=value, Upper=
    value, DisplayUnit='unit');
[out] size as Integer (Brief="brief", Default=value, Lower=value, Upper=
    value);
[out] ParName(size) as base (Brief="brief", Default=value, Lower=value,
    Upper=value, DisplayUnit='unit');
[out] ParName(dim1,dim2,...) as base (Brief="brief", Default=value, Lower=
    value, Upper=value, DisplayUnit='unit');
ParName as Switcher (Brief="brief", Valid=[ "value1", "value2", ... ], Default="
    value");
VARIABLES
[in | out] VarName;
[in | out] VarName as base;
[in | out] VarName as ModelBase [(Brief="brief")];
VarName as base (Brief="brief", Default=value, Lower=value, Upper=value,
    DisplayUnit='unit');
VarName(size) as base (Brief="brief", Default=value, Lower=value, Upper=
    value, DisplayUnit='unit');
VarName(dim1,dim2,...) as base (Brief="brief", Default=value, Lower=value,
    Upper=value, DisplayUnit='unit');
CONNECTIONS
varName<in> to varName<out>;
EQUATIONS
["equation name"] expression = expression;
INITIAL
["equation name"] varName = expression | value;
SET
ParName = <expression involving only parameters> | value;
size = < a integer value >;
end
```

---

The Optimization Entity

```
Optimization name [as base]
MINIMIZE
    expression1;
    expression2;
MAXIMIZE
    expression3;
    expression4;
EQUATIONS
    expression5 < expression6;
    expression7 > expression8;
FREE
    variable1;
    variable2;
end
```

The Estimation Entity

```
Estimation EstimationName as FlowSheetName
ESTIMATE
#PARAMETER  START      LOWER      UPPER      UNIT
ParName  StartValue  LowerValue  UpperValue  'unit'
EXPERIMENTS
# DATA FILE      WEIGHT
"FileName1.dat"    WeigthValue1;
"FileName2.dat"    WeigthValue2;
. . .
OPTIONS
. . .
end
```

The experimental data – FileName.dat

MEASURE	time	Param1	Param2	. . .	ParamN
[ UNIT	'timeUnit'	'unit1'	'unit2'	. . .	'unitN' ]
[ VARIANCE	timeVar	var1	var2	. . .	varN ]
DATA	Timevalue	value1	value2	. . .	valueN

Simple Equations

```
[ "equation name" ] expression = expression;
```

Expressions:

```
algebraic operations: +, -, *, /, ^
built-in functions: See list above
plugin Method: PluginName.PluginMethod(arguments)
```

Conditional Equations:

```
if LogicalExpression then
    [ "equation name" ] Equations;
else
    [ "equation name" ] Equations;
end

switch Valid_Switcher_ParName
case "validName":
    [ "equation name" ] Equations;
    [ when LogicalExpression switchto "Any validName" ];
. . .
end
```

LogicalExpressions:

```
Relational operations:
    <, >, <=, >=
Logical constants:
    true, false
Logical operations:
    and, or, not, equal
```

Iterative equations:

```
for index in [StartValue:EndValue]
    [ "equation name" ] Equations;
end
```

## Anexo I: Scripts de los modelos de equipos utilizados en la simulación y optimización de la producción de amoníaco.

```

Model splitter2
  ATTRIBUTES
    Pallete = true;
    Icon = "icon/splitter_column";
    Brief = "Splitter with 2 outlet streams";
    Info =
      "==" Assumptions ==
      *Thermodynamics equilibrium
      *Adiabatic

  == Specify ==
  * The inlet stream
  * One FlowRatios of split of the outlet streams:

      FlowRatios(i) = (Mole Flow of the outlet stream i /
                      Mole Flow of the inlet stream)
                      where i = 1, 2
  ";

  VARIABLES
    in Inlet as stream (Brief = "Inlet stream", PosX=0.5, PosY=0, Symbol="{in}");
    out Outlet1 as stream (Brief = "Outlet stream 1", PosX=0.25, PosY=1, Symbol="{out1}");
    out Outlet2 as stream (Brief = "Outlet stream 2", PosX=0.75, PosY=1, Symbol="{out2}");
    FlowRatios(2) as fraction (Brief = "Distribution of Outlets", Default=0.33, Symbol="{phi}");

  EQUATIONS

  "Normalize Flow Ratios"
    sum(FlowRatios) = 1;

  "Flow"
    Outlet1.F = Inlet.F * FlowRatios(1);
    Outlet1.F + Outlet2.F = Inlet.F;

  "Composition"
    Outlet1.z = Inlet.z;
    Outlet2.z = Inlet.z;

  "Pressure"
    Outlet1.P = Inlet.P;
    Outlet2.P = Inlet.P;

  "Enthalpy"
    Outlet1.h = Inlet.h;
    Outlet2.h = Inlet.h;

  "Temperature"
    Outlet1.T = Inlet.T;
    Outlet2.T = Inlet.T;

  "Vapourisation Fraction"
    Outlet1.v = Inlet.v;
    Outlet2.v = Inlet.v;

end

```

**Figura 55.** Script de la base de datos EMSO del modelo Splitter 2.

El modelo de separador de corrientes utilizado en el caso de estudio de la producción de amoníaco, es el descrito en la figura 55 (splitter 2). Al ser un modelo de la base de datos de EMSO en el script principal del caso de estudio (figura 21) solo se hace referencia al modelo splitter escribiendo: using “mixers\_splitters/splitter”. Como se puede observar el modelo requiere que se especifiquen las variables de la corriente de entrada y la relación de caudales entre las dos corrientes de salida. Además, este modelo supone que las condiciones en el separador son adiabáticas y que existe el equilibrio termodinámico dentro del mismo.

```

Model flash_steady

ATTRIBUTES
  Palette      = true;
  Icon         = "icon/Flash";
  Brief        = "Model of a static PH flash.";
  Info         =
    "This model is for using the flashPH routine available on VRTherm.

== ASSUMPTIONS ==
* perfect mixing of both phases;
* thermodynamics equilibrium.
* static model.

== SPECIFY ==
* The Inlet stream;
* The heat duty;
* The outlet pressure.
";

PARAMETERS

outer PP      as Plugin(Brief = "External Physical Properties", Type="PP");
outer NComp   as Integer;

VARIABLES

in Inlet      as stream      (Brief="Feed Stream", PosX=0, PosY=0.48, Symbol="_{in}");
out OutletLiquid as liquid_stream (Brief="Liquid outlet stream", PosX=0.43, PosY=1, Symbol="_{out}^{Liquid}");
out OutletVapour as vapour_stream (Brief="Vapour outlet stream", PosX=0.43, PosY=0, Symbol="_{out}^{Vapour}");
in InletQ      as power      (Brief="Heat Duty", PosX=1, PosY=0.81, Protected =true, Symbol="Q_{in}");

vfrac         as fraction    (Brief="Vaporization fraction", Symbol="\phi", Protected =true);
h             as enth_mol    (Brief="Mixture enthalpy", Hidden =true);
Pratio        as positive    (Brief = "Pressure Ratio", Symbol = "P_{ratio}", Protected =true);
Pdrop         as press_delta (Brief = "Pressure Drop", DisplayUnit = 'kPa', Symbol = "\Delta P", Protected =true);

EQUATIONS

if vfrac > 0 and vfrac <1
then
  "The flash calculation"
    [vfrac, OutletLiquid.z, OutletVapour.z] = PP.Flash(OutletVapour.T, OutletVapour.P, Inlet.z);
else
  "Chemical equilibrium"
    [vfrac, OutletLiquid.z, OutletVapour.z] = PP.FlashPH(OutletLiquid.P, h, Inlet.z);
end

"Global Molar Balance"
  Inlet.F = OutletVapour.F + OutletLiquid.F;

"Vapour Fraction"
  OutletVapour.F = Inlet.F * vfrac;

"Energy Balance"
  Inlet.F*(h - Inlet.h) = InletQ;
  Inlet.F*h = Inlet.F*(1-vfrac)*OutletLiquid.h + Inlet.F*vfrac*OutletVapour.h;

"Thermal Equilibrium"
  OutletVapour.T = OutletLiquid.T;

"Mechanical Equilibrium"
  OutletVapour.P = OutletLiquid.P;

"Pressure Drop"
  OutletLiquid.P = Inlet.P - Pdrop;

"Pressure Ratio"
  OutletLiquid.P = Inlet.P * Pratio;

end

```

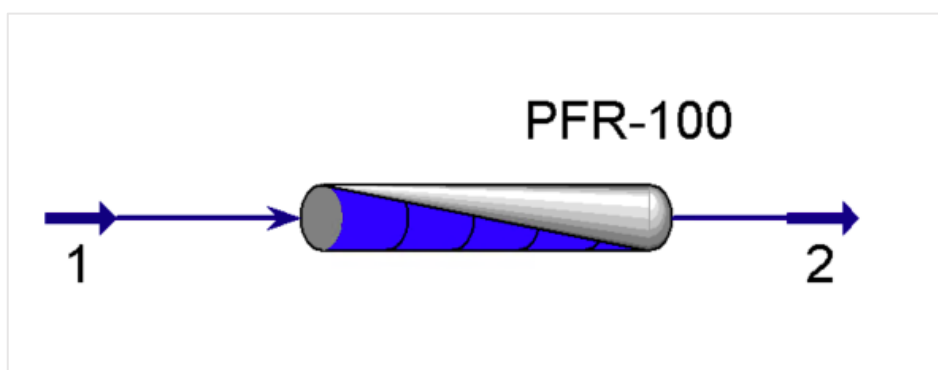
Figura 56. Script de la base de datos EMSO del modelo flash\_steady.

Para separar los componentes de salida del reactor en la producción de amoníaco se utilizan dos unidades flash. En la figura 56 se describe el modelo de flash utilizado donde es necesario especificar la corriente de entrada, el calor eliminado o aportado y la presión de salida del flash. Además, el modelo supone el estado estacionario, el equilibrio termodinámico de los componentes y la mezcla completa entre las fases líquido y vapor.

## Anexo II: Simulación de la producción de anhídrido acético en el simulador Aspen Hysys.

Aspen Hysys es un simulador de procesos secuencial-modular donde los cálculos del equipo se realizan de forma secuencial, el resultado de dejar un bloque se pasa como entrada al siguiente bloque, iterando a la solución de reciclaje. Se utiliza para modelar matemáticamente procesos químicos, desde operaciones unitarias hasta plantas químicas completas y refinerías. Además, Hysys se utiliza ampliamente en la industria y el mundo académico para la simulación dinámica y en estado estable, el diseño de procesos, el modelado de rendimiento y la optimización.

Para llevar a cabo la simulación de la producción de anhídrido acético, se ha optado por el fluid package Peng Robinson, una reacción tipo cinética donde se especifica la constante cinética, dato del problema. En la siguiente figura se observa el diagrama de procesos de la producción de anhídrido acético para este caso de estudio, constan únicamente de un reactor flujo pistón, una corriente de entrada y de salida. Al ser un reactor adiabático no existe un corriente de intercambio de energía.



*Figura 57. Diagrama del proceso de producción anhídrido acético en Hysys.*

Seguidamente, se puede observar el tipo de reacción que se ha optado en este caso, una reacción tipo cinética con el objetivo de poder introducir los datos en las constantes de reacción A y E. Despejando dichos valores de la cinética dada en el caso de estudio. Todo en concentraciones molares y en fase gas, debido a las temperaturas a la que se trabaja.

**Kinetic Reaction: Rxn-8**

**Stoichiometry and Rate Info**

Component	Mole Wt.	Stoich Coeff	Fwd Order	Rev Order
Acetone	58,080	-1,000	1,00	<empty>
Ketene	42,038	1,000	0,00	<empty>
Methane	16,043	1,000	0,00	<empty>
**Add Comp**				

**Basis**

Basis: **Molar Conc**  
Base Component: Acetone  
Rxn Phase: VapourPhase  
Min. Temperature: -273,1 C  
Max Temperature: 3000 C

Basis Units: **kgmole/m3**  
Rate Units: **kgmole/m3-h**

**Forward Reaction**

A: **2,9510e+018**  
E: **2,8473e+005**  
b: **0,00000**

**Reverse Reaction**

A': <empty>  
E': <empty>  
b': <empty>

**Equation Help**

$r = k \cdot f(\text{Basis}) - k' \cdot f'(\text{Basis})$   
 $k = A \cdot \exp(-E / RT) \cdot T^b$   
 $k' = A' \cdot \exp(-E' / RT) \cdot T^{b'}$   
T in Kelvin

Balance Error: **0,00000**  
Reaction Heat (25 C): **8,2e+04 kJ/kgmole**

Balance

Ready

Figura 58. Datos de la cinetica de la reacción en Hysys.

A continuación, se elige el reactor PFR (flujo pistón) y se añaden las corrientes de entrada y de salida. En el reactor se especifican las dimensiones, longitud y volumen. Tal y como se observa en la siguiente figura. Además, se especifica que es nula la perdida de presión y el intercambio de energía.

**Plug Flow Reactor: PFR-100 - Set-1**

Design Reactions Rating Worksheet Performance Dynamics

**Rating**

Sizing  
Nozzles

**Tube Dimensions**

Total Volume: **1,270 m3**  
Length: **2,280 m**  
Diameter: 0,8422 m  
Number of Tubes: **1**  
Wall Thickness: **5,000e-003 m**

**Tube Packing**

Void Fraction: **1,000**  
Void Volume: 1,270 m3

Delete OK Ignored

Figura 59. Dimensiones del reactor PFR en Hysys.

Finalmente, se obtienen los siguientes resultados expuestos en la figura 60 y las composiciones a la salida del reactor en fracciones molares, figura 61. Resultados que se han comentado en el apartado anterior 6.5. Comparación resultados en Matlab, Aspen Hysys y EMSO.

Plug Flow Reactor: PFR-100 - Set-1			
Design Reactions Rating Worksheet Performance Dynamics			
Worksheet	Name	1	2
Conditions	Vapour	1,0000	1,0000
Properties	Temperature [C]	761,9	664,0
Composition	Pressure [kPa]	162,1	162,1
PF Specs	Molar Flow [kgmole/h]	139,1	165,2
	Mass Flow [kg/h]	8000	8000
	Std Ideal Liq Vol Flow [m3/h]	10,18	11,28
	Molar Enthalpy [kJ/kgmole]	-1,223e+005	-1,030e+005
	Molar Entropy [kJ/kgmole-C]	269,4	242,0
	Heat Flow [kJ/h]	-1,702e+007	-1,702e+007
Delete OK Ignored			

Figura 60. Resultados de la simulación del PFR en Hysys.

Plug Flow Reactor: PFR-100 - Set-1			
Design Reactions Rating Worksheet Performance Dynamics			
Worksheet		1	2
Conditions	Acetone	0,9800	0,6671
Properties	Ketene	0,0100	0,1664
Composition	Methane	0,0100	0,1664
PF Specs			

Figura 61. Composiciones finales del PFR en fracciones molares en Hysys.



## 9. Bibliografía

- Caballero, J. A. (2018). *Simulación y optimización de los procesos químicos*. Alicante: Departamento de Ingeniería Química, Universidad de Alicante [apuntes de clase]
- Soares, R. d. (2002-2004). *EMSO Manual*. Rio Grande do Sul: UFRGS (Universidade Federal do Rio Grande do Sul).
- Soares, R. d. (s.f.). *ALSOC*. Obtenido de <http://www.enq.ufrgs.br/trac/alsoc/wiki>
- Dr. Rafael de Pelegrini Soares, D. G. (2001). *Modelagem e simulacao de processos usando o simulador EMSO (apuntes)*. Universidade Federal do Rio Grande do Sul.
- E. Tijssens a, D. R. (2002). Automatic differentiation for solving nonlinear partial differential equations: an efficient operator overloading approach. *Kluwer Academic Publishers*, 42.
- João Paulo Henrique1 | Ruy de Sousa Jr.1, 2. |. (2017). Optimization of chemical engineering problems with EMSO software . *WILEY*, 21.
- Leonardo Calabrez Gloeden Gonçalves, F. F. (2012). Implementation of Pareto Multi-objective Particle Swarm Optimization Algorithm in EMSO. *3rd International Conference on Engineering Optimization* (pág. 8). Rio de Janeiro, Brasil: EngOpt 2012.
- (FINAL), 2. C. (14 de Abril de 2021). *Chemical Engineering essentials for the CPI professional*. Obtenido de <https://www.chemengonline.com/2021-cepci-updates-february-prelim-and-january-final/>
- ODE45. (s.f.). *MathWorks*. Obtenido de <https://es.mathworks.com/help/matlab/ref/ode45.html>
- Turton, R. (2018). *Analysis, synthesis, and design of chemical processes, fifth edition*. Boston: Boston [etc.] : Prentice Hall .
- UU., D. d. (2018). *Instituto Nacional de Estándares y Tecnología (NIST)*. Obtenido de NIST: <https://webbook.nist.gov/cgi/cbook.cgi?Source=1997THE0B&Units=SI&Mask=1EFF>
- Yaws, C. L. (1997). *Handbook of chemical compound data for process safety*. Houston: Gulf Publishing Co.